NATIONAL TECHNICAL UNIVERSITY OF ATHENS
School of Electrical & Computer Engineering
Division of Communication, Electronic and Information Engineering

Knowledge extraction and Security in Internet of Things systems

Dissertation submitted for the degree of
Doctor of Philosophy
of

George C. Routis

Supervisor:
Assoc. Professor Ioanna Roussaki (NTUA)

Athens,
July 2024

This page was intentionally left blank.

NATIONAL TECHNICAL UNIVERSITY OF ATHENS
School of Electrical & Computer Engineering
Division of Communication, Electronic and Information Engineering

Knowledge extraction and Security in Internet of Things systems

Dissertation submitted for the degree of
Doctor of Philosophy
of

George C. Routis

Advisory Committee: Ioanna Roussaki
Miltiades Anagnostou
Symeon Papavassiliou

Approved by the seven-member committee on 8th July 2024.

Ioanna Roussaki Associate Professor (NTUA)	Miltiades Anagnostou Professor (NTUA)	Symeon Papavassiliou Professor (NTUA)
George Matsopoulos Professor (NTUA)	Athanasios Panagopoulos Professor (NTUA)	Eleni Stai Assistant Professor (NTUA)
Konstantinos Demestichas Assistant Professor (AUA)		

Athens, July 2024

.....
George C. Routis

Dr. Electrical and Computer Engineer, NTUA

Περίληψη

Είναι κάτι παραπάνω από προφανές ότι το Διαδίκτυο δεν είναι το ίδιο όπως πριν από μερικές δεκαετίες. Εξελίχθηκε και έφερε νέα αποτελέσματα και παράγοντες που το άλλαξαν και το έκαναν συμβατό με τις τρέχουσες ανάγκες. Έχουμε γίνει μάρτυρες μιας επανάστασης και της γέννησης τεχνολογιών όπως το Internet of Things, ευρέως γνωστό ως IoT στις μέρες μας. Το παραδοσιακό Διαδίκτυο έχει διεισδύσει σε πολλούς τομείς της καθημερινότητας και δεν ακολουθεί πλέον το αρχικό παράδειγμα, όπου ο χρήστης ανοίγει έναν επιτραπέζιο ή φορητό υπολογιστή και συνδέεται στο διαδίκτυο. Η ιδέα τώρα είναι ότι «αντικείμενα» όπως οι υπολογιστές μιας πλακέτας (SBC) ή οι μονάδες επεξεργασίας χαμηλού κόστους χρησιμοποιούνται σε τομείς όπως η Έξυπνη Γεωργία ή τα Οχήματα και μέσω ασύρματης σύνδεσης (2G/3G, 4G, 5G, Wi-Fi, LoRa , Zigbee, Sigfox, Bluetooth, Satellite,...) και σταθερές συνδέσεις (LAN, οπτικές ίνες,…) ο χρήστης μπορεί να αλληλεπιδράσει με αισθητήρες και ενεργοποιητές, προκειμένου να παρατηρήσει, μετρήσεις αρχείων καταγραφής δεδομένων και να ενεργήσει ανάλογα. Τα SBC και οι μονάδες επεξεργασίας χαμηλού κόστους είναι εξοπλισμένα με αισθητήρες, επομένως είναι ιδανικά για πολλές περιοχές όπου υπάρχει ανάγκη για ανίχνευση και καταγραφή δεδομένων σε μία ή περισσότερες συνδέσεις. Ωστόσο, η χρήση του IoT περιλαμβάνει πολλά περισσότερα από την απλή καταγραφή δεδομένων μέσω αισθητήρων. Οι μονάδες μπορούν να λειτουργούν ανεξάρτητα από την επίβλεψη του χρήστη. Μπορούν να λειτουργούν αυτόνομα μέσω της χρήσης προγραμμάτων που έχουν αποθηκευτεί σε αυτά. Για παράδειγμα, στο Internet of Vehicles (IoV) που είναι μια υποκατηγορία του IoT, οι κόμβοι (οχήματα) υπακούουν στη λογική του IoT. Επιπλέον, εκτελούν πιο σύνθετα προγράμματα, πραγματοποιούν αλλαγή της θέσης τους και αναλύουν το περιβάλλον για την ασφάλεια των χρηστών που βρίσκονται μέσα στα οχήματα. Το IoT μπορεί να διασυνδέεται με Cloud και διάφορες Υπηρεσίες Διαδικτύου, ώστε ένας χρήστης που ζει στη Γερμανία να μπορεί να ελέγχει τον έξυπνο μετρητή ενέργειας που είναι τοποθετημένος σε ένα σπίτι στην Ελλάδα.

Με την εξέλιξη της Μηχανικής Μάθησης (ML) είναι εφικτό να εκτελεστεί κώδικας ML σε SBC και να παρέχεται περισσότερη επεξεργαστική ισχύς στις συσκευές IoT. Δεν είναι ασυνήθιστο να συνδέεται μια κάμερα σε μια μονάδα επεξεργασίας χαμηλού κόστους και μέσω της χρήσης ενός εκπαιδευμένου μοντέλου ML εικόνας για να συμπεράνουμε ανάγκες επεξεργασίας ζωντανής εικόνας, όπως έλεγχος παρασίτων σε αγρόκτημα ή ανάλυση αλατότητας εδάφους και ανάλυση ασθενειών των φύλλων. Τα μη επανδρωμένα εναέρια οχήματα (UAV) συνδέονται ασύρματα με σταθμούς βάσης και μπορούν να τραβήξουν εικόνες από ένα αγρόκτημα προκειμένου να εντοπίσουν βασικά προβλήματα στο αγρόκτημα.

Αυτή η διατριβή τεκμηριώνει την τρέχουσα εργασία σε εφαρμογές του Διαδικτύου των Πραγμάτων στον πραγματικό κόσμο σε διαφορετικούς τομείς. Αρχικά, αναλύεται η λογική και οι διάφορες λεπτομέρειες του IoT και της Μηχανικής Μάθησης στη γεωργία ακριβείας. Εφαρμόστηκε ένα σχήμα για την ανίχνευση και αξιολόγηση διαφορετικών παραγόντων σε ένα εργαστηριακό πείραμα. Ανιχνεύεται και καταγράφεται η θερμοκρασία, η υπεριώδης ακτινοβολία, η υγρασία του εδάφους και η υγρασία του αέρα. Μέσω της χρήσης ενός εξελιγμένου επαναλαμβανόμενου νευρωνικού δικτύου - Μακροπρόθεσμης Μνήμης (RNN-LSTM), είναι σε θέση να προβλέπει καιρικές συνθήκες, ώστε ο χρήστης να μπορεί να εντοπίσει πότε χρειαζόταν άρδευση του φυτού ή του αγροκτήματος. Με αυτόν τον τρόπο ο χρήστης θα μπορούσε να εξοικονομήσει υδάτινους πόρους και χρήματα αποφεύγοντας την

περιττή/υπερβολική άρδευση. Η ενέργεια είναι ένας πολύτιμος / σπάνιος πόρος στα αγροκτήματα. Ως εκ τούτου, προχωρά και σε ανάλυση διαφορετικών μονάδων IoT και των σχετικών ασύρματων συστημάτων, προκειμένου να εντοπιστούν τρόποι βελτιστοποίησης της κατανάλωσης ενέργειας.

Δεύτερον, πραγματοποιήθηκαν πειράματα στη σφαίρα του Διαδικτύου των Οχημάτων (IoV), όπου η ασφάλεια είναι κρίσιμος παράγοντας. Πιο συγκεκριμένα, προσομοιώθηκε ένα δίκτυο οχημάτων IoV σε προσομοιωτή ns-3, όπου αναλύθηκαν διαφορετικά ασύμμετρα κρυπτογραφικά πρωτόκολλα (NTRU, ECC, HECC-g2, HECC-g3, RSA). Παρατηρήθηκαν μετρήσεις των χρόνων κρυπτογράφησης/αποκρυπτογράφησης, μεγέθη μηνυμάτων, χρόνους δημιουργίας υπογραφών, χρόνους επαλήθευσης υπογραφής, μεγέθη ανταλλαγής χειραψίας και χρόνους ανταλλαγής ψευδωνύμων, ενώ εξετάστηκε επίσης πώς επηρεάστηκε η ενέργεια των κόμβων (οχημάτων) κατά την εκτέλεση κάθε ασύμμετρου πρωτόκολλο.

Τρίτον, αναλύθηκαν τα αποτελέσματα των μοντέλων Μηχανικής Μάθησης και πιο συγκεκριμένα πώς συμπεριφέρεται το μοντέλο Συνελικτικού Νευρωνικού Δικτύου (CNN) όταν εκτελείται σε διαφορετικές αρχιτεκτονικές επεξεργασίας. Χρησιμοποιήθηκαν 3 SBC που ενσωμάτωσαν διαφορετικές μονάδες επεξεργασίας (CPU, GPU, TPU) που χρησιμοποιούνται στο κομμάτι του inference για την ανάλυση εικόνας που σχετίζεται με τις ασθένειες των φύλλων. Η τρέχουσα έρευνα επικεντρώθηκε κυρίως στη χρήση CPU, μνήμης RAM και swap, καθώς και στη θερμοκρασία και την κατανάλωση ενέργειας.

Τέταρτον, υπήρξαν εκτεταμένα πειράματα με μονάδες Arduino IoT σε φάρμες ρυζιού και καλαμποκιού, με χρήση Μηχανικής Μάθησης και πιο συγκεκριμένα τα CNN και τα RNN-LSTM. Η Γραμμική Παλινδρόμηση και η Πολλαπλή Παλινδρόμηση χρησιμοποιήθηκαν επίσης για την ανάλυση των αγροκτημάτων, και συγκεκριμένα σε αγροκτήματα ορυζώνων. Σχεδιάστηκε επίσης μια πρωτοποριακή συσκευή για ερευνητικούς λόγους, στα πλαίσια της παρούσας Διδακτορικής Διατριβής, στην οποία αναλύθηκε η συλλογή ρητίνης και καουτσούκ που υλοποιήθηκε με βάση τον μικροελεγκτή Arduino και διάφορους αισθητήρες, που μεταδίδει στον τελικό χρήστη πληροφορίες για την κατάσταση του περιβάλλοντος.

Τέλος, εφευρέθηκε ένα πρωτοποριακό γραμματοκιβώτιο για έντυπες επιστολές, με τη δυνατότητα να ενημερώνει τον χρήστη μέσω μηνυμάτων Short Message/Messaging Service (SMS) σε περίπτωση λήψης επιστολής. Χρησιμοποιεί έναν αισθητήρα υπέρυθρων που ανιχνεύει τη λήψη ενός νέου γράμματος για να αναγνωρίσει πότε υπάρχει ένα νέο γράμμα μέσα στο γραμματοκιβώτιο. Ενσωματώνει επίσης οθόνη υγρών κρυστάλλων (LCD) και πληκτρολόγιο για τον έλεγχο ορισμένων λειτουργιών όπως ο αριθμός κινητού τηλεφώνου του τελικού χρήστη, η τρέχουσα κατανάλωση και η ισχύς σήματος GSM/GPRS (Παγκόσμιο Σύστημα Κινητών Επικοινωνιών). Ο χρήστης μπορεί επίσης να ελέγξει τη διάρκεια ζωής της μπαταρίας. Η συσκευή έχει κατοχυρωθεί με δίπλωμα ευρεσιτεχνίας στον Οργανισμό Βιομηχανικής Ιδιοκτησίας Ελλάδος.

<u>Λέξεις Κλειδιά:</u> Τεχνητή Νοημοσύνη (ΤΝ), Συνελικτικά Νευρωνικά Δίκτυα (ΣΝΔ), Διαδίκτυο των Αντικειμένων (ΔτΑ), Επαναληπτικά Νευρωνικά Δίκτυα Μακράς-Βραχύχρονης Μνήμης,), Raspberry Pi, NVIDIA Jetson Nano, Google Coral TPU, Compute Unified Device Architecture (CUDA), Arduino, αισθητήρες, πρόβλεψη, Διαδίκτυο των Οχημάτων, Κρυπτογραφία, RSA, Ελλειπτικές Καμπύλες, Υπερ-Ελλειπτικές Καμπύλες, NTRU, AES.

Abstract

It is more than obvious that the Internet is not the same as it was a few decades ago. It has evolved and brought new results and factors that have changed it and made it compatible with the current needs. We have witnessed a revolution and the birth of technologies like the Internet of Things, widely known as IoT nowadays. The traditional Internet has penetrated many areas of our everyday life and we no longer follow the initial paradigm, where the user turns on a desktop or laptop PC and connects to the internet. The idea now is that "things" such as Single Board Computers (SBCs) or low-cost processing modules are being used in areas such as Smart Agriculture or Vehicles and via wireless (2G, 3G, 4G, 5G, Wi-Fi (Wireless Fidelity), LoRa (Long Range), Zigbee, Sigfox, Bluetooth, Satellite...) and stable connections (LAN – Local Area Network, optic fiber, …) the user can interact with sensors and actuators, in order to observe, data log measurements, and act accordingly. The SBCs and the low-cost processing modules are equipped with sensors, so they are ideal for many areas where there is need for sensing and datalogging over one or more connections. However, the usage of IoT involves so much more than just datalogging via sensors. Modules can work independently of user supervising. They can operate autonomously via the use of programs that have been stored on them. For instance, in the Internet of Vehicles (IoV) which is a sub-category of IoT, the nodes (vehicles) obey to the IoT rationale. Furthermore, they execute more complex programs, implement changing of their position, and analyze the environment for the safety of the users that are inside the vehicles. IoT can interface with Clouds and various Internet Services, so a user who lives in Germany can control the smart Energy Meter which is placed in a house in Greece.

With the evolution of Machine Learning (ML) it is feasible to execute ML code in SBCs and provide more processing power to the IoT devices. It is not uncommon to attach a camera to a low-cost processing unit and via the use of a trained image ML model to inference live image processing targeting needs such as pest control in a farm field or soil salinity analysis and leaf disease analysis. Unmanned aerial vehicles (UAVs) are connected wirelessly with base stations and can capture images from a farm field in order to identify essential problems in farm field.

This thesis documents our work on real world Internet of Things applications in different areas. First, we analysed the rationale and various details of IoT and Machine Learning in precision agriculture. A scheme was implemented in order to sense and evaluate different factors in a laboratory experiment. We sensed and logged temperature, Ultra Violet (UV) radiance, soil moisture and air humidity. Through the use of a sophisticated Recurrent Neural Network - Long Short Term Memory (RNN-LSTM), we were able to forecast weather conditions, so the user could identify when there was need to irrigate the plant or farm field (in cases of scaling up). This way the user could save water resources and money by avoiding unnecessary/excess irrigation. Energy is a valuable/scarce resource in farms, therefore we also proceeded to an analysis of different IoT modules and the related wireless systems, in order to identify ways to optimize energy consumption.

Secondly, we performed experiments within the realm of the Internet of Vehicles (IoV), where security is a critical factor. More precisely, we simulated an IoV network of vehicles in an ns-3 simulator, where different asymmetric cryptographic protocols Number Theory Research Unit (NTRU), Elliptic Curve Cryptography (ECC), Hyper Elliptic Curve Cryptography – genus 2 (HECC-g2),

Hyper Elliptic Curve Cryptography – genus 3 (HECC-g3), Rivest Shamir Adleman (RSA) were analysed. We observed metrics of encryption/decryption times, message sizes, signature generation times, signature verification times, exchange handshake sizes, and pseudonym exchange times, while we also examined how the energy of the nodes (vehicles) was affected when executing each asymmetric protocol.

Thirdly, we elaborated on the effects of Machine Learning models, and more precisely how the Convolutional Neural Network (CNN) model behaves when executed in different processing architectures. We used 3 SBCs that incorporated different processing units: Central Processing Unit (CPU), Graphics Processing Unit (GPU) Tensor Processing Unit (TPU) used in the inference part on image analysis related to leaves' diseases. Our research focused mainly on CPU-, Random Access Memory (RAM)-, and swap memory usage, as well as temperature and energy consumption.

Fourthly, we experimented extensively with Arduino IoT modules in rice and maize farms, in cooperation with Machine Learning and more specifically CNNs and RNN-LSTMs. Linear Regression and Multiple Regression were used for farm metrics' analysis, especially in rice fields farms. There is also a pioneer device analyzed towards resin and rubber collection presented based on Arduino microcontroller and various sensors, that transmits to the end user information about the environmental conditions of the resin/rubber collection via GSM/GPRS or via Xbee Zigbee.

Lastly, a pioneer mailbox for hardcopy letters, was invented, with the ability to inform the user via Short Message/Messaging Service (SMS) messages if a letter is received. It uses an InfraRed sensor which senses the reception of a new letter in order to identify when there is a new letter inside the mailbox. It also incorporates a Liquid Crystal Display (LCD) screen, and keypad in order to control some functions such as the end user's mobile number, the current consumption and the GSM/GPRS (Global System for Mobile Communications/General Packet Radio Service) signal strength. The user can also check the life of the battery. The device has been patented in the Hellenic Industrial Property Organisation ("OBI" in greek).

Keywords: Machine Learning (ML), Convolutional Neural Networks (CNN), Internet of Things (IoT), Recurrent Neural Networks – Long Short-Term Memory (RNN-LSTM), Raspberry Pi, NVIDIA Jetson Nano, Google Coral TPU, Compute Unified Device Architecture (CUDA), Arduino, sensors, forecasting, Internet of Vehicles (IoV), cryptography, RSA, ECC, HECC, NTRU, Advanced Encryption Standard (AES).

Σύνοψη

Στην παρούσα διδακτορική διατριβή παρουσιάζεται η έρευνα και τα πειράματα του υποψηφίου που έλαβαν χώρα σε θέματα Διαδικτύου των Αντικειμένων, εστιάζοντας κυρίως σε 2 άξονες, την Εξαγωγή Γνώσης και την Ασφάλεια.

Στο ***1ο κεφάλαιο*** γίνεται λόγος για τις εφαρμογές του Διαδικτύου των Αντικειμένων (ΔτΑ) στην Γεωργία Ακριβείας. Η Γεωργία διατηρεί μια εξέχουσα θέση στην κοινωνία και κατ' επέκταση στην κοινότητα σε παγκόσμια κλίμακα. Υπάρχουν πολλές δυσκολίες που πρέπει να ξεπεραστούν σε αυτόν τον τομέα, όπως για παράδειγμα η ασφάλεια του φαγητού, η σωστή χρήση των διαφόρων πόρων που υπάρχουν στην φύση, οι διακυμάνσεις του κλίματος, η αύξηση ζήτησης για φαγητό και η σπατάλη βιοποικιλότητας. Το Διαδίκτυο των Αντικειμένων, είναι μια τεχνολογία που μπορεί να βοηθήσει ώστε να δοθεί μια λύση σε πολλά από τα προαναφερθέντα προβλήματα, με την ενεργοποίηση της Γεωργίας Ακριβείας ως μέρος της Γεωργίας 4.0, φέρνοντας με αυτόν τον τρόπο τα καλύτερα αποτελέσματα σε κάθε περίπτωση. Μία βασική πρόκληση είναι ο μετριασμός της χρήσης νερού στην Γεωργία.

Το ΔτΑ πραγματοποιεί την σύνδεση ανάμεσα σε πολλές συσκευές, προκειμένου να αλληλοεπιδράσει με την ανταλλαγή δεδομένων και λαμβάνοντας σαν πλεονέκτημα υπηρεσίες Νέφους. Τα δεδομένα που συλλέγονται από διάφορες συσκευές ΔτΑ που βρίσκονται τοποθετημένες σε στρατηγικά σημεία, μπορούμε στην συνέχεια να τα επεξεργαστούμε και να τα αξιοποιήσουμε σε αναλύσεις και αποφάσεις σχετικές με βελτιστοποίηση. Η τεχνολογία ΔτΑ, έχει εισέλθει στον εμπορικό τομέα και υπάρχουν πολλές προσπάθειες προκειμένου να γίνει πιο φιλική προς τον χρήστη. Το ΔτΑ παρουσιάζει την δυνατότητα που έχει στο να βελτιώσει τις ζωές των ανθρώπων καθώς και στην βελτίωση διαφόρων λειτουργιών σε ποικίλους τομείς, όπως οι εξής: τομέας υγείας, τομέας εκπαίδευσης, τομέας παραγωγής και αγροτικός τομέας. Πιο συγκεκριμένα, η αποδοχή μιας τέτοιας τεχνολογίας, όπως το ΔτΑ στον τομέα της Γεωργίας οδηγεί προς την εμφάνιση της Γεωργίας 4.0. Μία από τις ουσιώδεις δυσκολίες που υπάρχουν σήμερα είναι συνδεδεμένη με το σύστημα διατροφής, καθώς σε παγκόσμιο επίπεδο υπάρχει η ανάγκη για αύξηση της παραγωγής φαγητού επιπλέον 50% μέχρι την χρονιά 2050, σε σχέση με το 2010 η παροχή τροφής σε έναν προβλεπόμενο πληθυσμό περίπου 10 δισεκατομμυρίων ανθρώπων είναι ιδιαίτερα δύσκολη, ενώ παράλληλα αυξάνονται οι πιεσμένοι και πεπερασμένοι πόροι (στη φύση) και η ανάγκη προσαρμογής στις ταχέως μεταβαλλόμενες κλιματικές συνθήκες. Ωστόσο, το σημερινό κλίμα καθώς και οι υπόλοιπες περιβαλλοντικές συνθήκες δεν βοηθούν στην απαιτούμενη αύξηση της φυτικής παραγωγής με την παραδοσιακή προσέγγιση στη Γεωργία. Οι τεχνολογίες που περιγράφονται στο κεφάλαιο 1 μπορούν να βοηθήσουν στην επίλυση αυτού του προβλήματος και να επιφέρουν αύξηση των αποδόσεων καθώς και μείωση των αναγκαίων πολύτιμων πόρων. Σε αυτό το σημείο, η Γεωργία Ακριβείας και η έξυπνη Γεωργία μπορούν να βοηθήσουν. Η υιοθέτηση των μεθόδων της ακριβούς άρδευσης βοηθά τους αγρότες να χρησιμοποιούν νερό μόνο στις καλλιέργειες που το χρειάζονται πραγματικά. Το αποτέλεσμα είναι η εξοικονόμηση υδάτινων πόρων.

Είναι γνωστό από την βιβλιογραφία, ότι οι Ηνωμένες Πολιτείες χρειάζονται περίπου το 80% νερό της χώρας και περισσότερο από το 90% που χρησιμοποιείται στις δυτικές πολιτείες. Στην Καλιφόρνια, και το έτος 2019, για να ποτίσουν 1.530.000 στρέμματα αμυγδάλου, οι αγρότες χρησιμοποίησαν 195,26 δισεκατομμύρια γαλόνια ετησίως. Σε μια άλλη εργασία, υποστηρίζεται

ότι η άνυδρη περιοχή όασης στο βορειοδυτικό τμήμα της Κίνας παρέχει νερό στο 95% του τοπικού πληθυσμού και στο 90% της οικονομικής περιουσίας λιγότερο από το 10% της κατεχόμενης περιοχής, που είναι η βασική ζώνη της περιοχής. Το σύστημα άρδευσης υποστηρίζει τη γεωργική και κοινωνικοοικονομική χρήση του νερού, και διατηρεί την ισορροπία του περιβάλλοντος, που καθορίζει την επιβίωση της όασης. Το σύστημα άρδευσης παρέχει υποστήριξη στους τομείς της Γεωργίας και του κοινωνικοοικονομικού τομέα. Για την περαιτέρω επέκτασή του, διατηρεί μια ισορροπία στο περιβάλλον που είναι υπεύθυνο για την επιβίωση του οικοσυστήματος της όασης. Τα τελευταία χρόνια, λόγω της δουλειάς των ανθρώπων και λόγω των κλιματολογικών αλλαγών, παρατηρείται ποσοτική μεταβολή του νερού που υπάρχει σε άνυδρες περιοχές (όαση). Συνέπεια αυτού είναι ότι επηρεάζονται οι υπόγειες δεξαμενές νερού, το νερό που απαιτείται για τη γεωργική δραστηριότητα και η αντίστοιχη αλατότητα του νερού. Σύμφωνα με την Παγκόσμια Τράπεζα, περίπου το 70% του πόσιμου νερού είναι απαραίτητο για τη Γεωργία, ενώ τα εργοστάσια και η υπόλοιπη βιομηχανία χρειάζονται περίπου το 20%. Το άλλο 10% του νερού χρησιμοποιείται διεθνώς για οικιακές εργασίες. Μέχρι το έτος 2050, οι άνθρωποι που ζουν στη Γη θα φτάσουν τον εκπληκτικό αριθμό των 10 δισεκατομμυρίων. Είναι προφανές ότι οι ανάγκες σε νερό και τρόφιμα θα αυξηθούν.

Ποιο είναι το πρόβλημα

Το IoT βρίσκεται σε πολύ ώριμο επίπεδο προκειμένου να εφαρμοστεί στον τομέα της Γεωργίας και να δώσει λύσεις σε πολλά ζητήματα, όπως η βιωσιμότητα, η ποιότητα και η ποσότητα στην απόδοση, η σχέση κόστους-αποτελεσματικότητας. Αναπτύσσονται συστήματα έξυπνης άρδευσης γύρω από συσκευές IoT, που αποτελούνται από αισθητήρες, CPUs, και ενεργοποιητές, που στοχεύουν στην εκτίμηση πολλών παραμέτρων, όπως η κατάσταση του εδάφους, η καλλιέργεια, τα καιρικά φαινόμενα και παρέχουν υποστήριξη σχετικά με αποφάσεις που λαμβάνονται για την άρδευση φυτών. Για το λόγο αυτό, θα πρέπει να υπάρχει σωστή εκτίμηση για το πόσο νερό πρέπει να χρησιμοποιείται σε ένα αγρόκτημα, διαφορετικά θα έχουμε έλλειψη ή περίσσευμα νερού, με αποτέλεσμα να υπάρχουν προβλήματα.

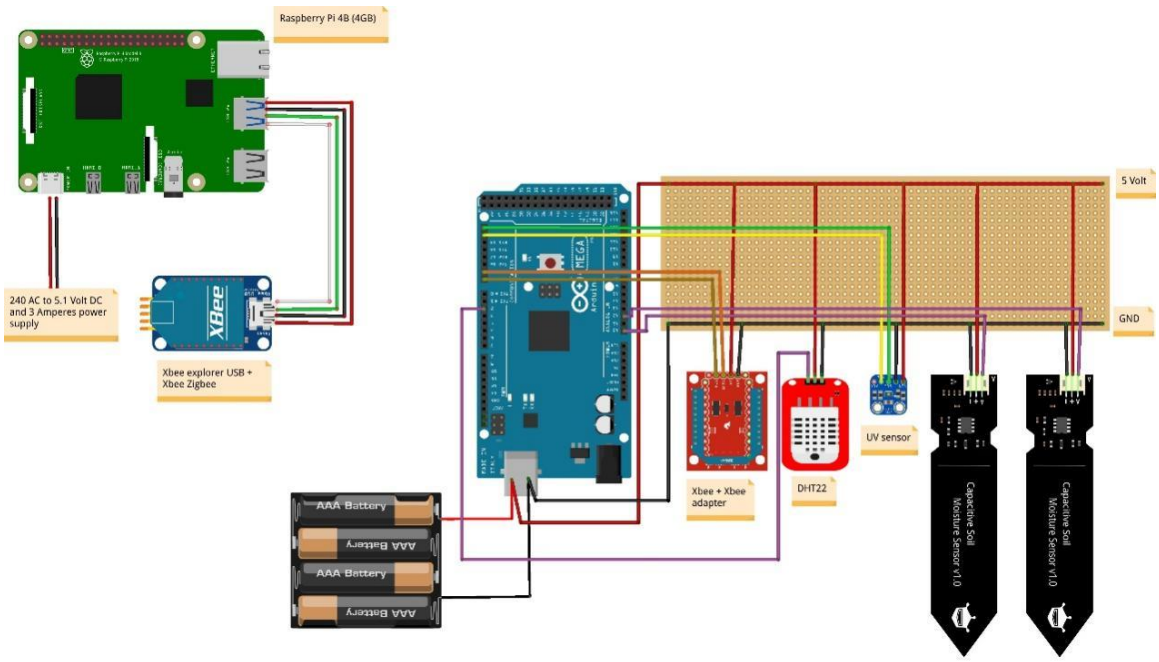
Λύση στο πρόβλημα

Για την λύση του παραπάνω προβλήματος χρησιμοποιήθηκε ο μικροελεγκτής Arduino σε συνδυασμό με τους κάτωθι αισθητήρες, όπως φαίνεται στην **Εικόνα 1**:

a) χωρητικός αισθητήρας για μέτρηση της υγρασίας εδάφους
b) ο DHT22 αισθητήρας για την μέτρηση της θερμοκρασίας και της υγρασίας
c) ο VEML6070 αισθητήρας για την μέτρηση της Υπεριώδους ακτινοβολίας

Στην συνέχεια, καταμετρήσαμε τις διάφορες παραμέτρους από τους αισθητήρες και πήραμε κάποια αποτελέσματα. Με την βοήθεια Μηχανικής Μάθησης και πιο συγκεκριμένα ενός νευρωνικού δικτύου RNN-LSTM, χρησιμοποιήσαμε σαν είσοδο τα δεδομένα που πήραμε από τους αισθητήρες και μπορέσαμε να κάνουμε προβλέψεις για μελλοντικές τιμές σε σχέση με την θερμοκρασία, υγρασία αέρα, υγρασία εδάφους και υπεριώδους ακτινοβολίας. Επίσης, για το συγκεκριμένο πείραμα χρησιμοποιήσαμε διαφορετικά συστήματα ασύρματης επικοινωνίας και καταγράψαμε την κατανάλωση ενέργειας με ειδική συσκευή καταμέτρησης ενέργειας που αναπτύχθηκε στο Εργαστήριο Διάχυτης Νοημοσύνης του Ε.Μ.Π. Τα αποτελέσματα φαίνονται στον **Πίνακας 1**.

.



Εικόνα 1 Η συνδεσμολογία του κυκλώματος για την μέτρηση των διαφόρων παραμέτρων του φυτού (θερμοκρασία αέρα, υγρασία αέρα, UV ακτινοβολία, υγρασία εδάφους).

	Ρεύμα Λειτουργίας (mA)									
Στοιχείο	**Διάταξη 1**	**Διάταξη 2**	**Διάταξη 3**	**Διάταξη 4**	**Διάταξη 5**	**Διάταξη 6**	**Διάταξη 7**	**Διάταξη 8**	**Διάταξη 9**	**Διάταξη 10**
Soil moisture sensor	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8
DHT22 sensor	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
VEML6070 UV sensor	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
Arduino MEGA2560 R3 (measured without pins used)	109	109	109	109	109					
Xbee Zigbee	41					41				
SIM900 GPRS (EGSM 900) mean of (PCL=5)		310					310			
SIM7600E 4G (20Mbps)			624					624		
Adafruit RFM96W LoRa Radio (+13 dBm)				51					51	
Adafruit RFM96W LoRa Radio (+20dBm)					152					152
Raspberry Pi 4B						290	290	290	290	290
Συνολικό ρεύμα κατανάλωσης (mA)	164.3	433.30	747.3	174.3	275.3	345.3	614.3	928.3	355.3	456.3
Τάση (Volts)	5	5	5	5	5	5	5	5	5	5
Κατανάλωση Ισχύος (in mWatts)	821.5	2166.5	3736.5	871.5	1376.5	1726.5	3071.5	4641.5	1776.5	2281.5

Πίνακας 1 Συγκριτική απεικόνιση των διαφόρων ασύρματων μονάδων με διαφορετικούς μικροελεγκτές, ως προς την κατανάλωση ενέργειας.

Το ***κεφάλαιο 2*** αναφέρεται στην χρήση διαφορετικών βαθμίδων επεξεργασίας, όπως CPU, GPU και TPU στον τομέα της Μηχανικής Μάθησης, καθότι είναι ευρέως συνδεδεμένη τα τελευταία χρόνια με μονάδες ΔτΑ, και τονίζονται τα πλεονεκτήματα και τα μειονεκτήματα κάθε προσέγγισης. Τα τελευταία χρόνια εκτός από την κλασσική βαθμίδα ενός πολύ-πύρηνου επεξεργαστή που χρησιμοποιείται για την αναγνώριση εικόνας στο κομμάτι του λεγόμενου inference, δηλαδή την λειτουργία που γίνεται η αναγνώριση (συμπέρασμα), έχουν κάνει την εμφάνισή τους και άλλες βαθμίδες, όπως πολύ- νηματικές κάρτες γραφικών, τα γνωστά Graphics Processing Units (GPUs). Επίσης, τα τελευταία χρόνια έχουν κάνει εμφάνιση και τα Tensor Processing Units (TPUs). Στο κεφάλαιο 2 παρουσιάζουμε πως ανταποκρίνονται 3 διαφορετικές αρχιτεκτονικές στο ίδιο εκπαιδευμένο μοντέλο αναγνώρισης εικόνας, κάτι που έχει μεγάλη σημασία για την αυτόματη αναγνώριση και κατηγοριοποίηση καταστάσεων από δεδομένα πραγματικού χρόνου. Συγκεκριμένα, το βασικό θέμα που εξετάζει το κεφάλαιο αυτό είναι η επεξεργασία εικόνας με τη σχετική ταξινόμηση εικόνων με φύλλα ανάλογα με το αν είναι άρρωστα ή υγιή. Προκειμένου να επεκταθεί ένα προηγούμενο σύνολο δεδομένων ταξινόμησης που αποτελείται από 10-15 κλάσεις, αποφασίστηκε να οριοθετηθεί το πρόβλημα με 33 κατηγορίες για τα επεξεργασμένα φύλλα, με μια προσπάθεια να μην καταστραφεί η ακρίβεια του μοντέλου. Το σύνολο δεδομένων που χρησιμοποιήθηκε συγκεντρώθηκε από αυτό ένα ειδικό ελεύθερης πρόσβασης αποθετήριο[1]. Το σύνολο δεδομένων που χρησιμοποιήθηκε δεν ήταν κατανεμημένο το ίδιο σε κάθε φάκελο- κλάση. Έγινε λοιπόν προ-επεξεργασία, προκειμένου να υπολογιστεί ο αρχικός αριθμός των εικόνων. Ο αριθμός των εικόνων στην τάξη ήταν 152. Σε κάθε τάξη διατηρήθηκαν 3 *min (όπου min ο ελάχιστος αριθμός εικόνων ανά κλάση = 152) = 456 εικόνες, ενώ υπάρχουν τάξεις με περισσότερες από 1000 εικόνες, που θα είχαν αρνητικές επιπτώσεις στην ακρίβεια του μοντέλου, αφού η φάση εκμάθησης θα ήταν προσαρμοσμένη σε αυτούς. Η προ-επεξεργασία πραγματοποιήθηκε στο Νέφος μέσω της βοηθητικής χρήσης της εφαρμογής Google Drive. Η εκπαίδευση όλου του σχήματος υλοποιήθηκε στο Google Colab, ένα εργαλείο που κάνει καλή χρήση της αρχιτεκτονικής GPU και TPU για να επιταχύνει τον κώδικα Μηχανικής Μάθησης, γραμμένο σε γλώσσα προγραμματισμού python. Στη συνέχεια, το εκπαιδευμένο μοντέλο μεταφορτώθηκε σε SBCs προκειμένου να εφαρμοστεί η πρόβλεψη σε άγνωστες (νέες) εικόνες. Οι αλγόριθμοι Μηχανικής Μάθησης εκτελέστηκαν σε Υπολογιστές Single Board τεχνολογίας ΔτΑ, όπως: Raspberry Pi 3B+, Raspberry Pi 4B, NVIDIA Jetson Nano, Google Coral TPU Edge Dev Board. Ο στόχος ήταν να παραχθεί ένα μοντέλο χρησιμοποιώντας όσο το δυνατόν λιγότερους πόρους, πιο συγκεκριμένα χαμηλή RAM και μειωμένη ισχύ CPU. Οι εικόνες φορτώνονταν κάθε φορά σε μοντέλο ML και εφαρμόστηκε ένα τυχαίο φιλτράρισμα σε αυτές για να αλλάξουν διάφορες παραμέτρους όπως το εύρος των χρωμάτων. σε τιμή εύρους 0-255, φωτεινότητα, ζουμ κ.λπ. Η ιδέα σε αυτές τις διαδικασίες ήταν να έχουμε όσο πιο ρεαλιστικό σύνολο δεδομένων είναι δυνατόν, επειδή οι εικόνες που θα τροφοδοτούνται από τον χρήστη δεν θα ήταν σε άριστη κατάσταση, και έτσι το μοντέλο ML θα έπρεπε να λάβει υπόψη διάφορες ατέλειες. Άρα, το μοντέλο θα έπρεπε να χειρίζεται περιπτώσεις όπου η εικόνα πχ περιστρέφεται ή δεν έχει τον κατάλληλο φωτισμό κ.λπ.

Στις συσκευές ΔτΑ είναι πολύ σημαντικό να καταναλώνεται όσο το δυνατόν λιγότερη ενέργεια, επειδή υπάρχουν περιορισμοί ισχύος, ειδικά εάν το Single Board Computer (SBC) λειτουργεί με την υποστήριξη μιας μπαταρίας μαζί με ένα ηλιακό πάνελ ή μια μικρή ανεμογεννήτρια. Εκτός από μετρήσεις που αφορούσαν RAM, CPU, θερμοκρασία και χρόνο, έγιναν και άλλες μετρήσεις

[1] https://github.com/spMohanty/PlantVillage-Dataset

σχετικά με την κατανάλωση ενέργειας. Χρησιμοποιήθηκε ειδική συσκευή μέτρησης που κατασκευάστηκε στο Εργαστήριο Διάχυτης Νοημοσύνης του ΕΜΠ. Καταγράφηκαν δεδομένα σχετικά με την καταγεγραμμένη τάση της συσκευής (σε Volts), το ρεύμα (σε χιλιοστά Αμπέρ) και την ισχύ (σε χιλιοστά Watt) του φορτίου.

Ποιο είναι το πρόβλημα

Εάν οι ασθένειες των φυτών δεν εντοπιστούν σε πρώιμο στάδιο, υπάρχει ο κίνδυνος αύξησης του κόστους παραγωγής στη Γεωργία. Αυτό δείχνει ότι θα πρέπει να υπάρχει ένα σύστημα παρακολούθησης με υψηλή συχνότητα για την ανίχνευση πρώιμων σημείων της νόσου, πριν η ασθένεια καλύψει όλα τα αγροτικά φυτά. Είναι προφανές ότι η παρακολούθηση ολόκληρου του αγροκτήματος είναι αρκετά δύσκολη. Ωστόσο, με τη σημερινή τεχνολογία και μέσω της χρήσης μοντέλων απομακρυσμένης παρακολούθησης και Μηχανική Μάθηση είναι κάτι που μπορεί να πραγματοποιηθεί. Το τρέχον κεφάλαιο παρουσιάζει την εκτέλεση αλγορίθμων Μηχανικής Εκμάθησης που εκτελούνται σε υπολογιστές SBC για την αναγνώριση φυτικών ασθενειών.

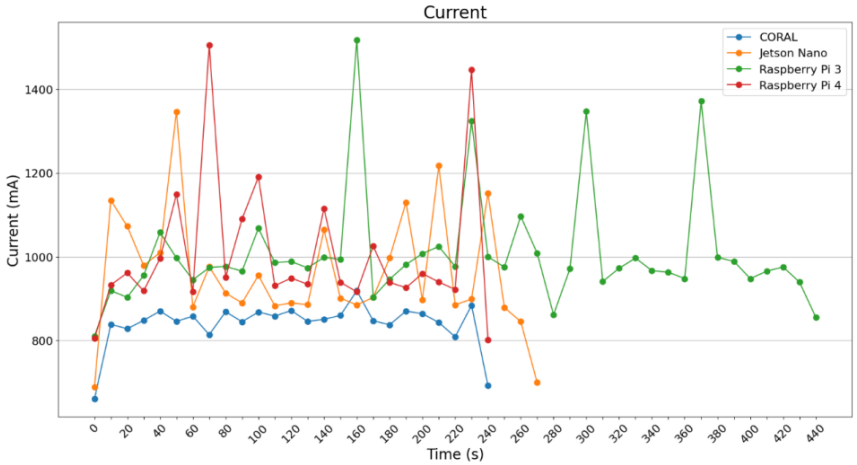
Τα τελευταία χρόνια, η Τεχνητή Νοημοσύνη έχει επιδείξει ιδιαίτερη αποτελεσματικότητα με τεράστιες εφαρμογές σε πολλούς τομείς, κάνοντας εντονότερη την ανάγκη για δεδομένα και πιο έξυπνους και πολύπλοκους αλγόριθμους επεξεργασίας. Αυτό το φαινόμενο υπογραμμίζει την ανάγκη για όσο το δυνατόν περισσότερο αποδοτική χρήση των διαθέσιμων πόρων, όπως RAM, CPU, ενέργεια. Μέσω της Μηχανικής Μάθησης οι μηχανές μπορούν να επεξεργάζονται διάφορες εργασίες αποφασίζοντας το αποτέλεσμα χωρίς ανθρώπινη αλληλεπίδραση, με βάση τη φυσική γνώση που τους παρέχουν οι άνθρωποι στα αρχικά στάδια. Η Τεχνητή Νοημοσύνη παρέχει υποστήριξη σε πολλά πεδία για την επίλυση προβλημάτων, για παράδειγμα: Μηχανική Μάθηση, επεξεργασία φυσικής γλώσσας (Natural Language Processing - NLP), επεξεργασία εικόνας και πολλά άλλα. Η Μηχανική Μάθηση είναι μια υποκατηγορία της Τεχνητής Νοημοσύνης. Αποτελείται από Αλγόριθμους που μπορούν να βελτιωθούν χωρίς ανθρώπινη παρέμβαση (αυτόματα) με βάση την εμπειρία. Το συγκεκριμένο κεφάλαιο χρησιμοποιεί την περίπτωση της εποπτευόμενης μάθησης, όπου ο χρήστης χρησιμοποιεί ετικέτες στα δεδομένα που τροφοδοτούνται στο μοντέλο ML. Το μοντέλο ML μπορεί να κατηγοριοποιήσει τις εισόδους και τις εξόδους δεδομένων. Το προτεινόμενο μοντέλο ML ελέγχει έναν αριθμό περιπτώσεων που αποτελούν μέρος συγκεκριμένων κατηγοριών και χρησιμοποιεί γνωστές ετικέτες για να προσδιορίσει σε ποια κατηγορία ανήκει μια πρόσφατη είσοδος. Έτσι, ο μηχανισμός εκπαιδεύεται με τρόπο ώστε να μπορεί να διαχωρίζει χαρακτηριστικά με βάση το σύνολο δεδομένων εκπαίδευσης, το οποίο αποτελείται από τα δεδομένα εισόδου. Στη συνέχεια, ένα σύνολο δεδομένων επικύρωσης τροφοδοτείται στο μοντέλο ML. Η σχέση μεταξύ των δεδομένων εισόδου και των ετικετών εξόδου είναι γνωστή, επομένως το μοντέλο ML είναι σε θέση να αξιολογήσει τη λειτουργία εκμάθησης.

Το κομμάτι της επαλήθευσης λειτουργεί ως εξής: τα δεδομένα επικύρωσης εισάγονται στο μοντέλο ML και συγκρίνονται με τις πραγματικές τιμές της εξόδου. Στην φάση εκμάθησης ο χρήστης τροφοδοτεί το μοντέλο ML με δοκιμαστικό σύνολο δεδομένων προκειμένου να έχει μια αξιολόγηση του πόσο ακριβής έχει γίνει ο μηχανισμός. Στην τελευταία φάση, ο χρήστης καλύπτει/κρύβει τις ετικέτες από το μοντέλο ML, ωστόσο, το μοντέλο ταξινομεί τα δεδομένα εισόδου με όσα έχει μάθει μέχρι τώρα. Στο τέλος της λειτουργίας, μπορεί να υπολογίσει τον αριθμό των περιπτώσεων που ταξινομήθηκαν σωστά και έτσι, το μοντέλο μπορεί να αξιολογηθεί ως προς την αξιοπιστία του. Σε όλα τα πειράματα χρησιμοποιήθηκαν τα Νευρωνικά Δίκτυα Συνέλιξης (CNN), μια λύση για εργασία με εικόνες και πιο συγκεκριμένα για προβλήματα

ταξινόμησης. Ταξινόμηση είναι η λειτουργία τροφοδοσίας του μοντέλου ML με εικόνες και το μοντέλο προκύπτει σε ποια κατηγορία ανήκει η εικόνα, υποδεικνύοντας ένα ποσοστό.

Στο κεφάλαιο αυτό ασχολούμαστε με τα εξής δεδομένα: χρήση CPU, ακρίβεια για κάθε κλάση, διάρκεια του inference για κάθε κλάση με χρήση του Google Colab, χρήση της μνήμης (%), χρήση της μνήμης σε Mbytes, θερμοκρασία, κατανάλωση ρεύματος, χρήση CPU (για batch size = 2, 4, 8, 16), χρήση μνήμης (για batch size = 2, 4, 8, 16), χρήση μνήμης σε Mbytes με χρήση ImageDataGenerator (για batch size = 2, 4, 8, 16), θερμοκρασία με χρήση ImageDataGenerator (για batch size = 2, 4, 8, 16), κατανάλωση ρεύματος με χρήση ImageDataGenerator (για batch size = 2, 4, 8, 16), χρήση CPU, χρήση RAM (%) για το Raspberry Pi 3B+, χρήση RAM (MBytes) για το Raspberry Pi 3B+, χρήση swap μνήμης (MBytes) για το Raspberry Pi 3B+, θερμοκρασία Raspberry Pi 3B+, CPU, χρήση RAM (%) για το Raspberry Pi 4, χρήση μνήμης RAM (%) για το Raspberry Pi 4, χρήση μνήμης RAM (MBytes) για το Raspberry Pi 4, θερμοκρασία Raspberry Pi 4.

Στην **Εικόνα 2** παρουσιάζεται η κατανάλωση ενέργειας στις διαφορετικές μονάδες που χρησιμοποιήσαμε για το πείραμα.



Εικόνα 2 Συγκριτική απεικόνιση της κατανάλωσης ενέργειας και των τεσσάρων SBCs (Single Board Computers).

Στο ***3ο κεφάλαιο*** γίνεται λόγος για την εφαρμογή μεθόδων υπολογισμού της αλατότητας του εδάφους σε καλλιέργειες ρυζιού, μέσω της χρήσης δορυφορικών εικόνων ή εικόνων από μη επανδρωμένα αεροχήματα (Unmanned Aerial Vehicles – UAVs / drones).

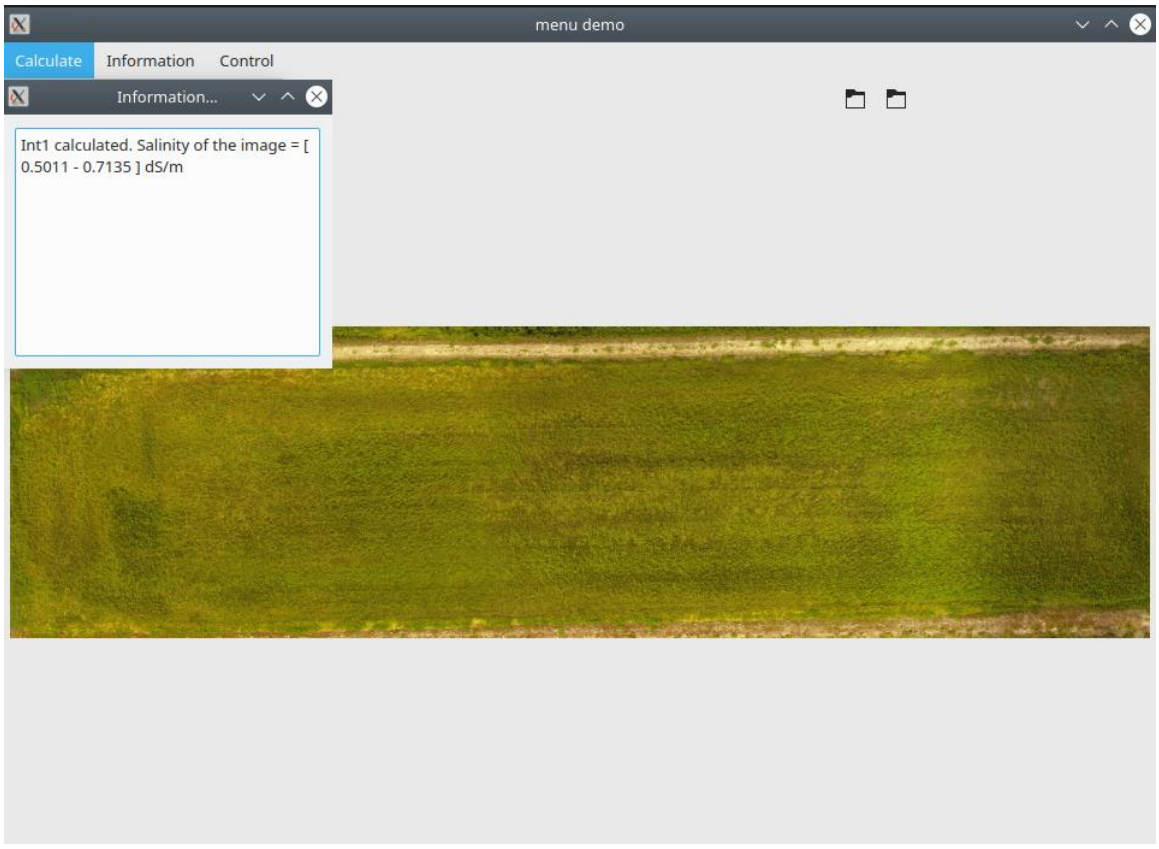
Μία από τις σημαντικότερες ανησυχίες στον αγροτικό τομέα είναι η σωστή χρήση των πόρων, για παράδειγμα: νερό, λιπάσματα, έδαφος. Αυτού του είδους οι πόροι σχετίζονται άμεσα με τα χρήματα, για κάθε μέσο αγρότη. Έτσι, όλοι προσπαθούν να χρησιμοποιούν τους πόρους αυτούς μόνο όταν και για όσο χρειάζεται. Έτσι αποφεύγεται η υπερβολική χρήση πόρων, καθώς και η χρήση μικρότερης ποσότητας λιπασμάτων και νερού από ό,τι χρειάζεται, κάτι που μπορεί να έχει αρνητικές συνέπειες στην απόδοση. Για να ληφθεί η σωστή απόφαση, χρειάζεται η χρήση κατάλληλων εργαλείων. Στο κεφάλαιο αυτό παρουσιάζεται μια εφαρμογή που δημιουργήθηκε με το σκεπτικό να βοηθήσει τους αγρότες να λάβουν μια εκτίμηση της αλατότητας που υπάρχει στο έδαφος των αγροκτημάτων τους μέσω της χρήσης μόνο εικόνων Unmanned Aerial Vehicle (UAV), χωρίς καμία χρήση αισθητήρων εδάφους ή οποιουδήποτε άλλου εξοπλισμού. Η κύρια ιδέα ήταν να δημιουργηθεί μια απλή διεπαφή για άτομα που δεν είναι ειδικευμένα στους υπολογιστές, ώστε να μπορούν να χρησιμοποιούν εύκολα την εικόνα που λαμβάνεται από το

UAV/drone για να λάβουν πληροφορίες για τη μέση τιμή αλατότητας σε ένα χωράφι στο αγρόκτημα ρυζιού.

Η εφαρμογή δημιουργήθηκε χρησιμοποιώντας script python και τις ακόλουθες βιβλιοθήκες/πακέτα: ConfigParser, Geospatial Data Abstraction Library (GDAL), matplotlib, numpy, opencv_python, Osgeo, Pandas, PyQt5, Rasterio, sklearn, Tifffile.

Χρησιμοποιεί επεξεργασία εικόνας ώστε να μπορεί να υπολογίσει τους διάφορους δείκτες βλάστησης εξετάζοντας τις ζώνες κάθε εικόνας και αξιολογώντας την αλατότητα μέσω της χρήσης ειδικών μαθηματικών μοντέλων.

Η διαδικασία είναι η εξής: ο χρήστης επιλέγει από το μενού «Υπολογισμός» ποιος Δείκτης Βλάστησης (Vegetation Index - VI) ταιριάζει στην κατάστασή του ανάλογα με την εμπειρία που έχει, γιατί κάθε VI χρησιμοποιεί διαφορετικές ζώνες. Αφού επιλεχθεί η εικόνα, ο χρήστης επιλέγει ποιον VI θα χρησιμοποιήσει για να υπολογίσει την μέση αλατότητα σε όλο το χωράφι. Σαν αποτέλεσμα η εφαρμογή τυπώνει ένα μήνυμα όπως φαίνεται στην **Εικόνα 3**, όπου φαίνεται ένα εύρος αλατότητας στο συγκεκριμένο χωράφι.

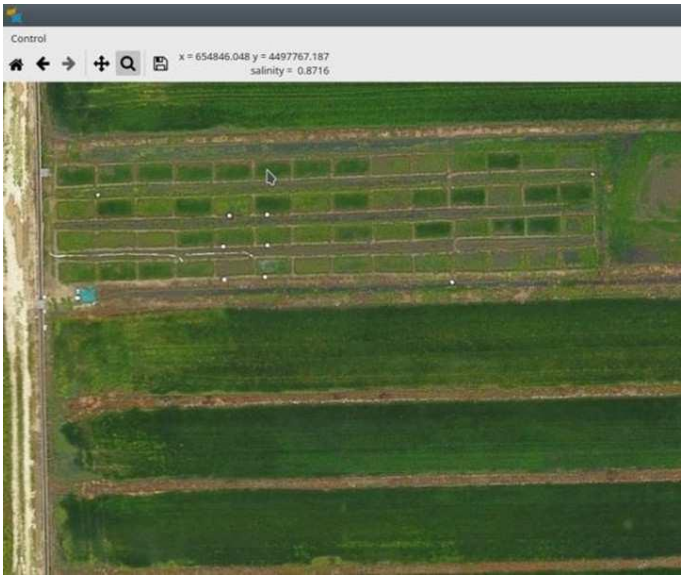


Εικόνα 3 Απεικόνιση της εφαρμογής, και του εύρους απεικόνισης αλατότητας εδάφους που εκτιμά η εφαρμογή.

Στο ***κεφάλαιο 4*** γίνεται αναφορά για την εφαρμογή της Μηχανικής Μάθησης σε συνδυασμό με το ΔτΑ στην Γεωργία. Η ανάμειξη του εδάφους με διαλυτά άλατα αφήνει το έδαφος αλατούχο.

Το τελευταίο είναι ένα σημαντικό πρόβλημα, επειδή η αλατότητα μετριάζει την παραγωγικότητα της γης. Κάποιες φορές τα αλατούχα εδάφη είναι αποτέλεσμα της άρδευσης, λόγω των αλάτων που περιέχει το νερό. Τα αλατούχα εδάφη θα μπορούσαν να είναι το αποτέλεσμα της αυξημένης χρήσης νερού σε κοντινά παράκτια πεδία, λόγω της διείσδυσης της θάλασσας και των πλημμυρών που σημειώνονται κοντά σε αυτές τις περιοχές ως αποτέλεσμα των καταιγίδων των μεσογειακών περιοχών. Στα δέλτα των ποταμών, εντός της ευρωμεσογειακής περιοχής, η κύρια καλλιέργεια που καλλιεργείται είναι το ρύζι. Ως αποτέλεσμα του γεγονότος ότι η αλατότητα είναι ενδημική σε χωράφια κοντά στις ακτές, τα φυτά ρυζιού πρέπει να γεμίσουν με γλυκό νερό για να μετριαστεί η αλατότητα του νερού. Έτσι, απαιτείται πολύ μεγάλη ποσότητα νερού για τη μείωση της αλατότητας και σημαντική ποσότητα ενέργειας για την άντληση νερού από τα ποτάμια.

Ένας τρόπος για τη συνεχή δειγματοληψία της αλατότητας, είναι μέσω αισθητήρων IoT, που τοποθετούνται στο έδαφος και ανανεώνουν το νερό στο αγρόκτημα μόνο όταν είναι απαραίτητο. Ωστόσο, όταν ο ορυζώνας είναι τεράστιος, όπως συχνά συμβαίνει σε πραγματικές καταστάσεις, είναι πολύ ακριβό για τον τελικό χρήστη ή τον αγρότη να τοποθετήσει πολλούς αισθητήρες ΔτΑ στο αγρόκτημά του. Επιπλέον, μπορεί να υπάρχει πρόβλημα όταν πρέπει να τοποθετηθούν γεωργικά μηχανήματα στα αγροτεμάχια, όπως τρακτέρ. Για το λόγο αυτό, είναι επιτακτική η ανάγκη παρακολούθησης της συγκέντρωσης αλατιού στο χωράφι χωρίς αισθητήρες τοποθετημένους στο έδαφος, αλλά έμμεσα, μέσω επεξεργασίας UAV και δορυφορικής εικόνας. Έτσι, όπως αναφέρθηκε προηγουμένως, μια (ακριβή) λύση για τη μέτρηση της αλατότητας του εδάφους θα μπορούσε να είναι η τοποθέτηση πολλών αισθητήρων ΔτΑ σε πολλά σημεία μέσα στο αγρόκτημα ρυζιού. Μια (μη δαπανηρή) λύση θα ήταν η ακόλουθη: εάν τα φυτά σε μια καλλιέργεια ρυζιού καλύπτονται με αυξημένη τιμή αλατότητας που μετράται από έναν αισθητήρα ΔτΑ, τότε όλα τα φυτά κοντά στο ρύζι θα αντιμετωπίζουν το ίδιο στρες αλατότητας του εδάφους. Αυτό μπορεί να αναλυθεί από UAV ή δορυφορικές εικόνες, εξ αποστάσεως, χωρίς αισθητήρες.



Εικόνα 4 Υπολογισμός αλατότητας εδάφους όπου φαίνονται και οι συντεταγμένες, ανάλογα με την επιλογή του σημείου που τοποθετείται ο δείκτης του ποντικιού.

Η **Εικόνα 4** απεικονίζει ένα παράδειγμα ενός Ενεργοποιητή Βέλτιστης Ποιότητας Νερού. Οι χρήστες εισάγουν μια εικόνα UAV που τραβήχτηκε από drone. Στη συνέχεια, επιλέγουν το χρονικό εύρος που πρέπει να είναι σχετικό με την εικόνα που λαμβάνεται και οι αλγόριθμοι ταιριάζουν τη χρονική σήμανση της λήψης εικόνας με τις μετρήσεις από τους αισθητήρες εδάφους. Ο χρήστης επιλέγει έναν από τους 2 αισθητήρες ΔτΑ προκειμένου να χρησιμοποιηθεί ως κύρια αναφορά για τη σύνδεση εικόνας. Αυτό που κάνει ο Ενεργοποιητής (Enabler) είναι να εξάγει μια τιμή στην εφαρμογή με τις συντεταγμένες LONGITUDE, LATITUDE με τη σχετική εκτίμηση αλατότητας του εδάφους, ανάλογα με το πού τοποθετεί τον δείκτη του ποντικιού. Η εφαρμογή χρησιμοποιεί μεθόδους Αντίστροφη Στάθμιση Απόστασης (Inverse Distance Weighting - IDW)[2] και τον σχετικό κόμβο αισθητήρα IoT που τοποθετείται στο έδαφος.

Χρησιμοποιώντας αυτού του είδους την εκτίμηση της αλατότητας του εδάφους, ο αγρότης ή ο τελικός χρήστης είναι σε θέση να λάβει αποφάσεις σχετικά με το πότε θα τοποθετήσει νερό στο αγρόκτημα ρυζιού του, ακόμη και σε περιπτώσεις όπου το χωράφι δεν είναι εξοπλισμένο με αισθητήρα IoT. Το αγρόκτημα ρυζιού που απεικονίζεται στην **Εικόνα 4** περιέχει μόνο δύο αισθητήρες IoT για τη μέτρηση της αλατότητας του εδάφους. Επίσης, ο αγρότης αποφεύγει ζημιές στην απόδοση της καλλιέργειας του γιατί ενημερώνεται πολύ γρήγορα για την αύξηση της αλατότητας του εδάφους και εξοικονομεί πολύτιμο νερό που σε αυτές τις συνθήκες και για την ποσότητα που το χρειάζεται κοστίζει πολύ.

Παρακάτω θα αναφερθούν κάποια πειράματα που έγιναν στο πεδίο της Ευφυούς Γεωργίας με χρήση Μηχανικής Μάθησης και αισθητήρες ΔτΑ.

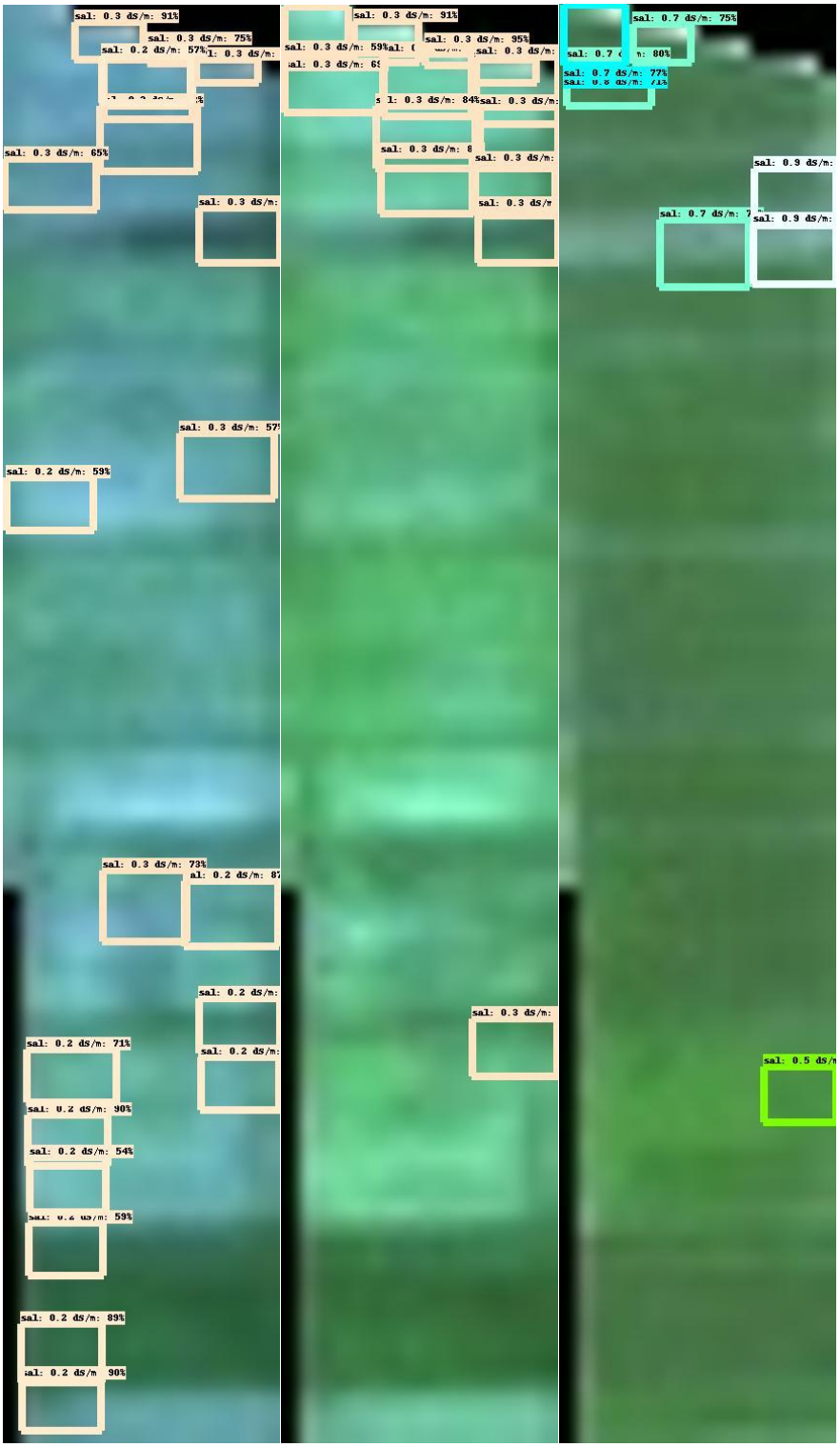
Στο 1ο πείραμα, παρουσιάζονται τα αποτελέσματα από την χρήση Μηχανικής Μάθησης, για την πρόβλεψη αγροτικών και μετεωρολογικών δεδομένων, με στόχο την πρόβλεψη καιρικών δεδομένων σε τοπικό επίπεδο. Συγκεκριμένα χρησιμοποιήθηκε το νευρωνικό δίκτυο RNN-LSTM το οποίο αφού τροφοδοτηθεί με χρονοσειρές δεδομένων, όπως η θερμοκρασία, η υγρασία αέρα, και η υγρασία κοντά στο φυτό, μπορεί να εξάγει συμπεράσματα για τις τιμές που θα λάβει κάθε μία από τις προηγούμενες 3 παραμέτρους στο μέλλον. Με αυτόν τον τρόπο ο γεωργός μπορεί να πάρει αποφάσεις με βάση προβλέψεις που εντοπίζονται σχετικά κοντά στο δικό του χωράφι και όχι γενικές, όπως συμβαίνει με την πρόβλεψη καιρικών δεδομένων για νομούς μιας χώρας ή ακόμα και ολόκληρη τη χώρα.

Σε άλλο πείραμα σε μια άλλη εφαρμογή Μηχανικής Μάθησης στο πεδίο της Γεωργίας, χρησιμοποιήθηκε ένα άλλο είδος νευρωνικού δικτύου, το λεγόμενο CNN, το οποίο εφαρμόζεται κυρίως στην αναγνώριση εικόνας. Η όλη υλοποίηση βασίστηκε σε δορυφορικές εικόνες που απεικονίζουν χωράφια ρυζιού και μετρήσεις που συγκεντρώθηκαν από αισθητήρες IoT που τοποθετήθηκαν μέσα στο αγροτεμάχιο. Οι αισθητήρες IoT συγκέντρωσαν πληροφορίες σε συνεχή συχνότητα, ανιχνεύοντας την αλατότητα του εδάφους.

Ουσιαστικά υπήρχε μια σύνδεση μεταξύ της ημερομηνίας λήψης των δορυφορικών εικόνων και των μετρήσεων των επίγειων αισθητήρων IoT. Διατηρήσαμε τις τιμές των αισθητήρων αλατότητας εδάφους IoT που τοποθετήθηκαν στο αγρόκτημα ρυζιού. Κάθε μέρα της λήψης δορυφορικής εικόνας συνδέθηκε με τη μέση αλατότητα που ανιχνεύτηκε από τις συσκευές IoT. Συνεπώς, κάθε τιμή αλατότητας στρογγυλοποιήθηκε σε μια τιμή. Κάθε νέα (στρογγυλοποιημένη)

[2] https://en.wikipedia.org/wiki/Inverse_distance_weighting

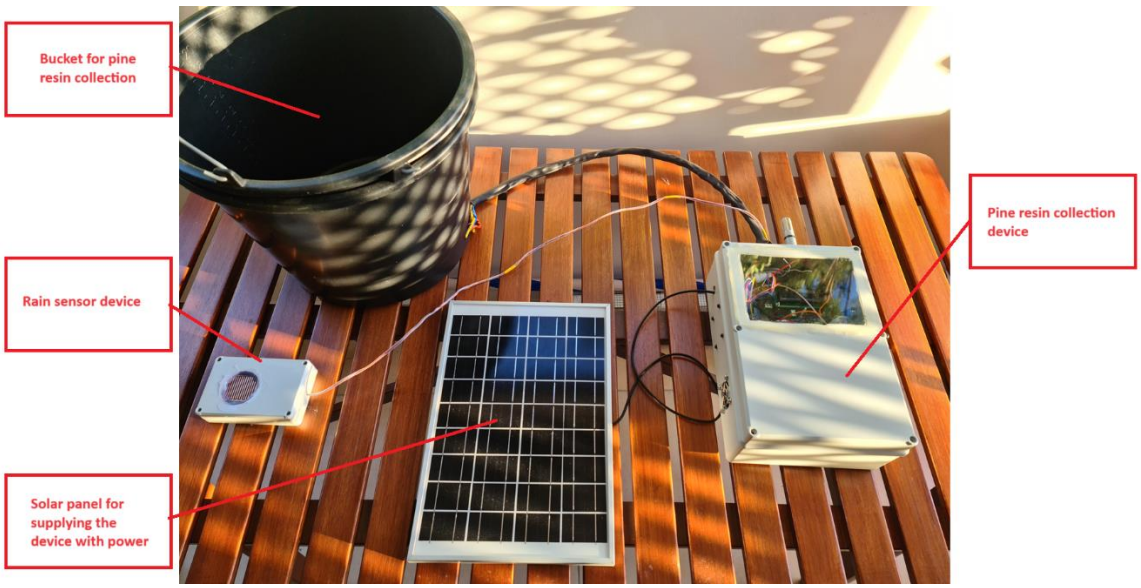
τιμή διατηρεί ένα εύρος ± 0,05. Για παράδειγμα, η φωτογραφία που τραβήχτηκε στις 12-07-2021 έχει αλατότητα 0,487 και μετά τη στρογγυλοποίηση η νέα της τιμή είναι 0,5 ± 0,05. Η βασική σκέψη ήταν να χρησιμοποιήσουμε μοντέλα Μηχανικής Μάθησης και ακριβέστερα CNN, ώστε να μπορούμε να εκπαιδεύσουμε σωστά το μοντέλο και κάθε φορά που ο τελικός χρήστης εισάγει μια εικόνα από μια δορυφορική πηγή ή μια πηγή UAV για να υποδείξει τις σχετικές περιοχές με αλατότητα, την τιμή της αλατότητας στην τονισμένη περιοχή και το σκορ εκτίμησης (ακρίβεια). Η ακολουθούμενη διαδικασία της τρέχουσας ιδέας είναι ευρέως γνωστή ως ταξινόμηση. Όπως είναι λογικό, δεν θα μπορούσαμε να έχουμε έναν αριθμό τάξεων που να σχετίζονται με κάθε δεκαδικό αριθμό αλατότητας, γιατί θα είχαμε έναν τεράστιο αριθμό κλάσεων. Αυτός ήταν ο λόγος που επιλέξαμε τη λύση στρογγυλοποίησης της τιμής της αλατότητας και αντιπροσωπεύουμε κάθε στρογγυλεμένο αριθμό με ένα εύρος, ώστε να έχουμε έναν λογικό αριθμό κλάσεων. Χρησιμοποιήσαμε 8 διαφορετικές τάξεις. Τα αποτελέσματα φαίνονται στην παρακάτω **Εικόνα 5**.



Εικόνα 5 Αποτέλεσμα υπολογισμού της αλατότητας εδάφους με χρήση Συνελικτικών Νευρωνικών Δικτύων (CNN).

Στο ***5ο κεφάλαιο*** γίνεται λόγος για μια ΔτΑ συσκευή ρητινοσυλλογής, πιο συγκεκριμένα ρητίνης πεύκου (pine resin). Γίνεται αναφορά στην σπουδαιότητα της ρητίνης παγκοσμίως και

παρουσιάζονται κάποιες από τις εφαρμογές που έχει η ρητίνη στην καθημερινότητά μας. Ένας από τους μεγάλους παγκόσμιους εξαγωγείς ρητίνης είναι η Κίνα, με 200.000 τόνους ετησίως. Γίνεται εκτενής αναφορά στην υπάρχουσα βιβλιογραφία και συγκεκριμένα στις υπάρχουσες μεθόδους συλλογής ρητίνης (tapping) που είναι οι εξής δύο: 1) τεχνική quarre και 2) τεχνική drill. Και στις 2 μεθόδους η ρητίνη συλλέγεται σε ένα δοχείο στο κατώτερο μέρος του δέντρου. Κατά την ανασκόπηση της σχετικής βιβλιογραφίας (έτος 2023) δεν βρέθηκε κάποια συσκευή ΔτΑ που να παρέχει στο συλλέκτη ρητίνης πληροφορίες τόσο για την συλλεχθείσα ποσότητα ρητίνης, όσο και περιβαλλοντικές πληροφορίες που χρειάζεται να έχει σε γνώση ο γεωργός. Για τον λόγο αυτόν κατασκευάστηκε και παρουσιάζεται μια καινοτόμος συσκευή συσκευής ρητίνης με κατάλληλα ηλεκτρονικά στοιχεία, που ενημερώνει τον γεωργό μέσω του ασύρματου πρωτοκόλλου Zigbee ή με SMS στο κινητό του για την ποσότητα ρητίνης η οποία έχει συλλεχθεί στο δοχείο καθώς και για λοιπές περιβαλλοντικές συνθήκες που οφείλει να γνωρίζει για την ποιότητα της ρητίνης. Υπάρχει η δυνατότητα να ενημερωθεί ο γεωργός με μήνυμα στο κινητό του για την υγρασία και θερμοκρασία αέρα καθώς και αν υπάρχει βροχόπτωση ή όχι στην περιοχή. Υπάρχει η δυνατότητα να πληροφορηθεί για όλα τα δεδομένα του μετρητή μέσω ενός push button και μιας οθόνη LCD όταν βρίσκεται δίπλα στην συσκευή, χωρίς να δαπανάται πολύτιμη ενέργεια για να αποσταλούν ασύρματα οι μετρήσεις. Επίσης, η συσκευή ΔτΑ διαθέτει μηχανισμό ψύξης όλου του συστήματος μέσω ψηκτρών και ενός μικρού ανεμιστήρα, ο οποίος ενεργοποιείται όταν η θερμοκρασία στο εσωτερικό της συσκευής αυξηθεί. Όλο το σύστημα τροφοδοτείται από ηλιακό πάνελ, το οποίο φορτίζει ένα ηλεκτροσυσσωρευτή («μπαταρία») μέσω κατάλληλου φορτιστή. Η όλη κατασκευή φαίνεται στην **Εικόνα 6**.



Εικόνα 6 Παρουσίαση της συσκευής ρητινοσυλλογής. Φαίνονται οι εξής βαθμίδες: το καλάθι συλλογής και καταμέτρησης της ρητίνης, ο αισθητήρας βροχής, το φωτοβολταϊκό πάνελ και η κεντρική μονάδα με τον μικροελεγκτή, τον φορτιστή, τον ηλεκτροσυσσωρευτή και τους επιμέρους αισθητήρες.

Στο ***6ο κεφάλαιο*** γίνεται λόγος για την διασφάλιση Ιδιωτικότητας σε περιβάλλον Διαδικτύου των Οχημάτων μέσω κρυπτογραφικών αλγορίθμων, όπως των μη συμμετρικών RSA, ECC και NTRU, και του συμμετρικού AES.

Η τεχνολογία IoT μπορεί να εφαρμοστεί στις μεταφορές, όπου η ενσωμάτωση του IoT παρουσιάζει έντονες προκλήσεις και ευκαιρίες στον εμπορικό τομέα. Η χρήση του IoT στον τομέα των μεταφορών μπορεί να οδηγήσει στην πρόληψη ατυχημάτων και να βελτιώσει θέματα όπως: κυκλοφοριακή συμφόρηση, διαχείριση κυκλοφορίας, καλύτερος προγραμματισμός. Μπορεί επίσης να βελτιώσει τους οδηγούς σε τομείς όπως: έξυπνη πλοήγηση, αυτόματη πληρωμή κατά την είσοδο στην περιοχή διοδίων. Με την εφαρμογή των βασικών αρχών του IoT στα οχήματα, υπάρχει η διευθέτηση ενός δικτύου οχημάτων και η ιδέα του περίφημου Διαδικτύου Οχημάτων, μια πολύ απαιτητική περιοχή που έχει συνέπειες στους ανθρώπους (οδηγούς).

Στο IoV υπάρχει μεγάλη ανάγκη για υψηλή ασφάλεια επειδή εμπλέκονται ανθρώπινες ζωές. Επομένως, η ασφάλεια έναντι των κινδύνων, εσωτερικού, εξωτερικού επιτιθέμενου ή και των δύο, είναι πολύ σημαντική. Από αυτή την άποψη, η διασφάλιση δεδομένων που προέρχονται από διαφορετικούς ενδιαφερόμενους φορείς εντός του πεδίου IoV γίνεται πολύ κρίσιμη και απαιτητική. Επιπλέον, ακολουθώντας τις τρέχουσες οδηγίες σχετικά με το απόρρητο των δεδομένων, όπως συμβαίνει σε όλο τον κόσμο, πολλοί ερευνητές εξετάζουν μεθόδους στον τομέα της ιδιωτικής ζωής δεδομένων, που περιέχουν την ιδέα IoV.

Τα οχήματα στον τομέα IoV ανταλλάσσουν δεδομένα μεταξύ τους και με τις Οδικές Μονάδες (RSU) εφαρμόζοντας τα Δίκτυα Ad hoc Οχημάτων (VANETs). Τα VANET έχουν εφαρμογή στα ευφυή συστήματα μεταφορών Intelligent Transportation System (ITS), διευκολύνοντας την πρόβλεψη στατικού πλούτου ή πλούτου δυναμικής ευφυΐας στα διάφορα ενδιαφερόμενα μέρη, όπως για παράδειγμα πληροφορίες που σχετίζονται με: ασφάλεια, λεπτομέρειες οδών, κόμβους, τοπολογία συχνά μεταβαλλόμενου πλαισίου. Οι ζωές των επιβατών ή των οδηγών μπορεί να υποστούν βλάβη όταν αποστέλλονται τροποποιημένα δεδομένα στο VANET. Επομένως, είναι πολύ σημαντικό να εφαρμοστεί η ασφάλεια και η προστασία δεδομένων στα δεδομένα όταν μεταδίδονται ή λαμβάνονται μέσω δικτύων IoV.

Αυτό που πραγματεύεται το τρέχον κεφάλαιο είναι η εκτίμηση πολλών πρωτοκόλλων που σχετίζονται με την ασύμμετρη κρυπτογράφηση και αποκρυπτογράφηση, για παράδειγμα. ECC, RSA, NTRU, υλοποιούνται σε τοπολογίες IoV, πραγματοποιώντας διαφορετικές μετρήσεις απόδοσης, όπως π.χ. ο κρυπτογραφικός αλγόριθμος AES είναι το κύριο σχήμα για όλες τις μετρήσεις εκτός από τα διαφορετικά ασύμμετρα πρωτόκολλα, που αναφέρθηκαν παραπάνω. Έτσι, οι διαφορετικές παράμετροι που μετρήθηκαν, εκτός από τον ασύμμετρο αλγόριθμο, ήταν το μέγεθος των μηνυμάτων, η ανταλλαγή ψευδωνύμων, ο τρόπος με τον οποίο πραγματοποιούνται τα νέα ψευδώνυμα με τη σχετική κατανάλωση ενέργειας. Η αξιολόγηση των προαναφερθέντων αξιολογήθηκε σε προσομοίωση μέσω της χρήσης λογισμικού ανοιχτού κώδικα ns-3 και Simulation of Urban Mobility (SUMO), λαμβάνοντας υπόψη θέματα όπως π.χ. CPU, κατανάλωση ενέργειας, RAM σε περιβάλλον IoT, όπου έχουν περιορισμένους πόρους.

Η έρευνα με ζητήματα απορρήτου στο IoV αποτελεί μια πρόκληση λόγω του γεγονότος ότι οι παθητικές και ενεργητικές επιθέσεις στοχεύουν στην ανάκτηση προσωπικών πληροφοριών. Όταν βρίσκεται σε εξέλιξη μια παθητική επίθεση, ο εισβολέας αναλύει τα δημόσια δεδομένα προκειμένου να εντοπίσει ευαίσθητες πληροφορίες. Ωστόσο, στην περίπτωση της ενεργητικής επίθεσης, ο εισβολέας στοχεύει στην πρόσβαση σε ιδιωτικές πληροφορίες για να τις αλλοιώσει. Για να δώσουμε ένα παράδειγμα, όταν υπάρχει επίθεση αλλοίωσης δεδομένων, ο εισβολέας προσπαθεί να εισαγάγει ή να αλλάξει τακτικά δεδομένα. Αυτό έχει επιπτώσεις στην απόδοση της εκπαίδευσης του εφαρμοσμένου αλγόριθμου Μηχανικής Μάθησης, για παράδειγμα ενός FDIA (False Data Injection Attacks).

Η βασική ιδέα του τρέχοντος κεφαλαίου είναι η ανάλυση μιας αποτελεσματικής μεθόδου που διατηρεί το απόρρητο τοποθεσίας σε μια υποδομή δικτύου IoV. Σύμφωνα με τα ευρήματα της αξιολόγησης, η μέθοδος που παρουσιάζει το τρέχον κεφάλαιο, έχει τροποποιηθεί προκειμένου να επιτευχθεί καλύτερη απόδοση σε μετρήσεις όπως το μέγεθος των μηνυμάτων, η χρονική διάρκεια που καταναλώνει κάθε μήνυμα και η κατανάλωση ενέργειας. Οι κρυπτογραφικοί αλγόριθμοι που χρησιμοποιούνται σε όλα τα πειράματα αναλύονται επίσης εκτενώς.

Παρουσιάζεται ένα νέο σχήμα για τη διατήρηση του απορρήτου τοποθεσίας στα δίκτυα IoV και το οποίο βασίζεται στην λογική MixGroup. Πολλά μηνύματα που αποστέλλονται εντός της περιόδου λειτουργίας του MixGroup περιλαμβάνουν κρυπτογράφηση σε δεδομένα έτσι ώστε να υπάρχει ακεραιότητα των δεδομένων, αναγνωριστικά των μεταδιδόμενων οντοτήτων και προστασία από επιτιθέμενους (περισσότερο γνωστούς ως υποκλοπείς). Η επιλογή του σωστού πρωτόκολλου κρυπτογράφησης είναι πολύ κρίσιμη, όπως μπορεί να καταλάβει κάποιος, γιατί επηρεάζει την αποτελεσματικότητα του μοντέλου και πρέπει να συμπεριφέρεται όσο καλύτερα μπορεί προκειμένου να βελτιώσει τις απαιτήσεις ασφαλείας που έχουν οριστεί. Υπάρχει ανάγκη για γρήγορη απόκριση μέσω συσκευών με περιορισμένους πόρους σε δίκτυα IoV, επομένως, ένας σοβαρός μηχανισμός θα πρέπει να δαπανά όσο το δυνατόν λιγότερους υπολογισμούς και ενεργειακούς πόρους και ταυτόχρονα να ελαχιστοποιεί το μέγεθος των δεδομένων που πρόκειται να υποβληθούν σε επεξεργασία. Στο προτεινόμενο σχήμα λαμβάνονται υπόψη τα ακόλουθα:

1. Δημιουργία κλειδιών: διαδικασία δημιουργίας κλειδιού, τη δημιουργία πιστοποιητικών, τη διάρκεια της κρυπτογράφησης/αποκρυπτογράφησης, τους χρόνους υπογραφής στη δημιουργία ή την επαλήθευση και άλλα ζητήματα.
2. Κρυπτογράφηση/αποκρυπτογράφηση μηνυμάτων
3. Παραγωγή και επαλήθευση ψηφιακών υπογραφών
4. Παραγωγή και επαλήθευση ασφαλείας

Η ιδέα του Mix-Group βασίζεται στα εξής:

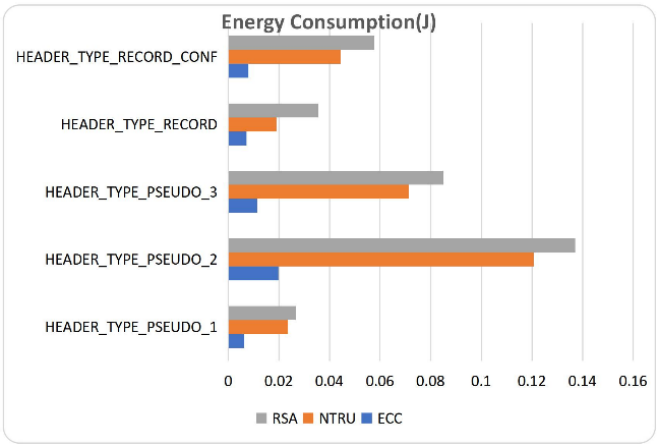
1. Μικρός αριθμός οχημάτων συγκεντρώνονται στα καθολικά κοινωνικά σημεία (global social spots) καθώς τα περισσότερα οχήματα συναντώνται σε διαφορετικά σημεία καθώς κινούνται στο οδικό δίκτυο.
2. Η πλειοψηφία των οχημάτων διαθέτουν κοινωνικά σημεία (individual social spots) στα οποία συναντάνε τα περισσότερα άλλα οχήματα μέσα στην ίδια μέρα.

Έχοντας υπόψη τα προαναφερθέντα, τα κοινωνικά σημεία χωρίζονται σε δύο κατηγορίες. παγκόσμια και προσωπικά. Έτσι, για να διαχειριστούμε όσο το δυνατόν περισσότερο το απόρρητο, υπάρχει η ανάγκη να κάνουμε καλή χρήση των δύο βασικών τεχνικών. Ο μηχανισμός που βασίζεται στο σκεπτικό του Mix-Group στοχεύει τόσο σε παγκόσμιο όσο και σε προσωπικό στην πορεία ενός οχήματος, ώστε να μπορεί να δημιουργήσει μια κάλυψη (τόπος) όπου υπάρχει ανταλλαγή ψευδωνύμων. Εκεί, ένας κόμβος (όχημα) μπορεί να ανταλλάσσει συνεχώς ψευδώνυμα προκειμένου να φτάσει στην καλύτερη μυστικότητα που μπορεί για την ταυτότητά του. Οι κόμβοι (οχήματα) που βρίσκονται εντός της περιοχής μετατρέπονται σε μέλη μιας ομάδας και χρησιμοποιούν κοινή ταυτότητα για την επικοινωνία με τους υπόλοιπους κόμβους και τον σχετικό μηχανισμό εφόσον αλλάζουν θέσεις εντός της στοχευόμενης περιοχής. Έτσι,

αξιοποιούν καλά και των δύο ειδών κοινωνικές θέσεις, καθώς υπάρχουν στο εκτεταμένο πεδίο και αποτελούν επαληθευμένα σημεία ανταλλαγής ψευδωνύμων.

Πολλές μετρήσεις έλαβαν χώρα στον εξομοιωτή ns-3 ακολουθώντας το μοντέλο του Mix-Group, και για τα 3 ασύμμετρα πρωτόκολλα: ECC, NTRU και RSA. Πιο συγκεκριμένα μετρήσεις έγιναν στις εξής παραμέτρους: χρόνος παραγωγής κλειδιού, χρόνος παραγωγής πιστοποιητικού, χρόνος κρυπτογράφησης, χρόνος αποκρυπτογράφησης, χρόνος παραγωγής υπογραφής, χρόνος πιστοποίησης υπογραφής, μέγεθος ανταλλαγής χειραψίας, μέγεθος ανταλλαγής ψευδωνύμων, μέγεθος ενεργοποίησης ψευδωνύμου, κατανάλωση ενέργειας, εντροπία.

Μία μέτρηση από τα αποτελέσματα που πήραμε από τις εξομοιώσεις στο λογισμικό ns-3 παρουσιάζονται παρακάτω, στην **Εικόνα 7**. Στην εικόνα αυτή παρουσιάζεται η κατανάλωση ενέργειας για τα διάφορα είδη μηνυμάτων που ανταλλάσσονται στο δίκτυο IoV



Εικόνα 7 Συγκριτική απεικόνιση της κατανάλωση ηλεκτρικής ενέργειας για τα διαφορετικά μηνύματα και για διαφορετικό πρωτόκολλο ασύμμετρης κρυπτογράφησης.

Στο ***7ο κεφάλαιο*** γίνεται αναφορά σε κρυπτογραφικές προσεγγίσεις Ελλειπτικών και Υπερελλειπτικών Καμπυλών για την προστασία της Ιδιωτικότητας σε περιβάλλον Διαδικτύου των Οχημάτων.

Το Διαδίκτυο των Αντικειμένων, είναι ένα δίκτυο που περιέχει φυσικές συσκευές, οχήματα ή άλλα στοιχεία που φέρουν αισθητήρες ή λογισμικό προκειμένου να στείλουν τις πληροφορίες, τα δεδομένα τους στο Διαδίκτυο. Μια υποκατηγορία του IoT είναι το IoV (Internet of Vehicles), το οποίο δίνει τη δυνατότητα στα αυτοκίνητα και στην υπόλοιπη οδική υποδομή να συνδέονται με το Διαδίκτυο και να στέλνουν τις πληροφορίες τους σε πολλούς αποδέκτες. Η επικοινωνία V2V μεταξύ των αυτοκινήτων και μεταξύ των αυτοκινήτων και των υπόλοιπων συσκευών, όπως η υποδομή (V2I) είναι πολύ μειωμένη στις μέρες μας. Μέσω της ισχυρής εξέλιξης του IoT και τις δυνατότητές του που διατηρεί για τη βελτίωση της εμπειρίας που αντιμετωπίζουν οι χρήστες σε καθημερινή βάση, μπορεί να επεκταθεί σε περιβάλλον IoV. Η σύγχρονη τεχνολογία εγγυάται αυτοματισμό, αποτελεσματική αξιοποίηση και ευκολία στον τομέα των μεταφορών, με απόλυτη προτεραιότητα στην εγγύηση της ασφάλειας των οδηγών και το μετριασμό του αριθμού των

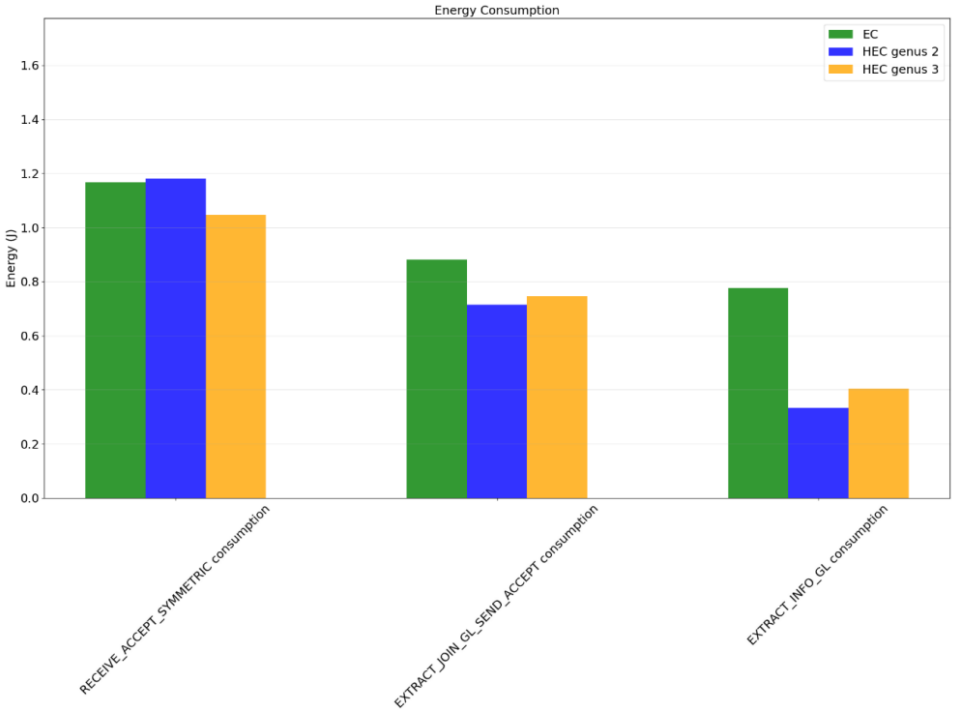
τροχαίων ατυχημάτων στους δρόμους που έχει ως αποτέλεσμα αυξημένο ποσοστό θανάτων στις μέρες μας. Καθώς εξελίσσονται οι τεχνολογίες 5G και 6G, προσφέρουν πολύ υψηλές ταχύτητες Διαδικτύου και μπορούν να επιτρέψουν μια επανάσταση στον τομέα του IoV.

Αν και υπάρχει μια ταχεία εξέλιξη που προκαλεί νέους κινδύνους στην ιδιωτική ζωή των χρηστών, υπάρχουν νέοι τρόποι στόχευσης για να διεισδύσουν στις προσωπικές πληροφορίες των χρηστών και να τις εκμεταλλευτούν. Πιο συγκεκριμένα στον τομέα της επικοινωνίας IoV (V2V/V2I) η ασφάλεια χρειάζεται προσεκτική παρακολούθηση, γιατί οι εισβολείς εκτός από την κλοπή προσωπικών δεδομένων μπορούν να θέσουν σε κίνδυνο και τη ζωή των οδηγών. Έτσι, για το λόγο αυτό, τα τελευταία χρόνια υπάρχει η ανάγκη της δημιουργίας ασφαλέστερου περιβάλλοντος επικοινωνίας, το οποίο πρέπει να προσαρμοστεί στο περιορισμένο περιβάλλον. Για τον λόγο αυτό, οι λύσεις ασφαλείας αυτών των ειδικών περιβαλλόντων και γενικά όταν σχετίζονται με το IoT, πρέπει να είναι αποτελεσματικές, γρήγορες και να μην φορτώνουν το δίκτυο με άχρηστη κίνηση δεδομένων.

Για τη μέτρηση του χρόνου χρησιμοποιήσαμε τη συνάρτηση: chrono::high_resolution_clock::now() που τοποθετείται στην αρχή και στο τέλος κάθε υπολογισμού. Οι μετρήσεις που ελήφθησαν είναι οι εξής: χρόνος δημιουργίας ζεύγους κλειδιού, χρόνος δημιουργίας πιστοποιητικού, χρόνος λήψης ιδιωτικού κλειδιού πιστοποιητικού, χρόνος εξαγωγής δημοσίου κλειδιού πιστοποιητικού, χρόνος κρυπτογράφησης μηνύματος, χρόνος αποκρυπτογράφησης μηνύματος, χρόνος δημιουργίας υπογραφής, χρόνος πιστοποίησης υπογραφής, χρόνος κωδικοποίησης, χρόνος αποκωδικοποίησης, σύγκριση μεγεθών μηνυμάτων (GL_LEADERSHIP_PROOF, VEHICLE_SEND_JOIN_GL, GL_ACCEPT, VEHICLE_INFORM), κατανάλωση ενέργειας.

Για την προσομοίωση χρησιμοποιήθηκαν τα ακόλουθα στοιχεία: Virtual Machine (VM) Linux Ubuntu 20.04.6 σε κεντρικό υπολογιστή Windows 10, 8 Giga Byte (GB) RAM, 4 πυρήνες (Intel Core i5-10300H, χρονισμένος @ 2,50 GHz), NS-3,30, SUMO 1.16.0 και Number Theory Library (NTL) 5.5.

Κάποια από τα αποτελέσματα που πήραμε φαίνονται παρακάτω στην **Εικόνα 8**:



Εικόνα 8 Συγκριτική απεικόνιση της διαφορετικής κατανάλωσης ηλεκτρικής ενέργειας για διαφορετικά μηνύματα και διαφορετικό αλγόριθμο ασύμμετρης κρυπτογράφησης.

Λέξεις Κλειδιά: Διαδίκτυο των Αντικειμένων, Συμμετρική Κρυπτογραφία, Ασύμμετρη Κρυπτογραφία, Μηχανική Μάθηση, Συνελικτικά Νευρωνικά Δίκτυα, Ανατροφοδοτούμενα Νευρωνικά Δίκτυα, Μικροελεγκτές, Ασύρματη δικτύωση, Ελλειπτικές Καμπύλες, Υπερ-Ελλειπτικές Καμπύλες, Εξαγωγή Γνώσης

This page was intentionally left blank.

In memory of my grandfather,
Georgios Routis

This page was intentionally left blank.

Contents

Σύνοψη 9
List of Figures 33
List of Tables 39
List of Abbreviations 41
Preface 45
Acknowledgements 45
Structure 46
Part 1: Knowledge Extraction in Internet of Things 47
Chapter 1: Internet of Things technology in precision agriculture 48
1.1 Introduction 48
1.2 Related State of the Art 49
1.2.1 Smart irrigation approaches focusing on IoT and ML 49
1.2.2 Current research towards energy consumption control 51
1.3 Analysis of the problem and the related proposed solution 52
1.3.1 What is the problem 52
1.3.2 Proposed solution to the problem 52
1.4 Evaluation of the experiments that took place 55
1.5 Experiments with the Arduino and its connected sensors 58
1.6 Savitzky-Golay Filtering 61
1.7 Introduction to RNN-LSTM neural networks 62
1.7.1 Recurrent Neural Network model 62
1.7.2 Long Short-Term Memory Module 64
1.8 Forecasting basil pot conditions using RNN-LSTM 66
1.9 Issues related to power consumption, control and monitoring 71
1.10 Conclusions 73
Chapter 2: Plant diseases identification using Single Board Computers (CPU, GPU, TPU) and Machine Learning models 76
2.1 Introduction 76
2.2 Related state-of-the-art 77
2.2.1 Machine Learning models used in the agri field 77
2.2.2 Single Board Computers executing CNN ML code 79
2.3 The description of the problem 80
2.4 Proposed Solution 81

2.5 The basics of Machine Learning ... 85
2.5.1 Convolution ... 86
2.5.2 Fully connected Layer ... 88
2.5.3 Softmax functions and ReLU ... 89
2.5.4. Forward and Backward computations ... 90
2.5.5 Connection between Machine Learning and agriculture ... 90
2.5.6 Classification of images ... 91
2.5.7 Analysis of the various experiments with the use of SBCs ... 92
2.5.8 Limitations and Risks in the Current Work ... 115
2.5.9 General evaluation results of the experiments ... 115
2.6 Conclusions ... 116
Chapter 3: An easy-to-use application based on evaluating ground soil salinity through the use of UAV images, without using more devices. A use case in Rice farm fields ... 119
3.1 Introduction ... 119
3.2 Related literature ... 119
3.3 Introduction to linear regression and multiple linear regression ... 121
3.3.1 Linear Regression basics ... 121
3.3.2 Multiple Regression basics ... 122
3.4 Evaluation and analysis of the proposed application ... 122
3.5 Data gathering and analysis ... 124
3.6 Image Analysis ... 127
3.7 Single Regression Analysis ... 129
3.8 Further general Analysis ... 130
3.9 Conclusions ... 135
Chapter 4: Machine Learning and IoT in the agri-domain ... 137
4.1 Introduction ... 137
4.2 Related State of Art literature ... 137
4.3 Rice water quality enabler ... 138
4.4 Maize irrigation Enabler ... 139
4.5 Optimal water quality via the use of Convolutional Neural Network ... 140
4.6 Conlusions ... 146
Chapter 5: An IoT device for monitoring resin/rubber collection from pine/rubber trees ... 149
5.1 Introduction ... 149

5.2 Existing literature 150
5.3 Proposed device for collecting resin or rubber from trees 152
5.4 Results and Conclusion 165
Part 2: Security in Internet of Things 167
Chapter 6: Cryptographic protocols in Internet of Vehicles fields 168
6.1 Introduction 168
6.2 State of the Art related to IoV privacy 169
6.3 Robustness and time complexity of the algorithmic schemes that were used 171
6.3.1 Consequences of the key lengths in the security level 173
6.4 Comparison between modern and classical asymmetric algorithmic schemes 174
6.5 Proposed Scheme 176
6.6 Evaluation of the experimental results 180
6.7 The novelty that the current chapter provides in the literature 187
6.8 Conclusions 188
APPENDIX A 189
A.1. RSA pseudocode 189
A.2. El Gamal pseudocode 189
A.3. El Gamal modified pseudocode 190
A.4. ECC pseudocode 191
A.5. HQC pseudocode 191
Chapter 7: Use of cryptographic techniques of Hyperelliptic and Elliptic Curves in order to improve Privacy in Internet of Vehicles 193
7.1 Introduction 193
7.2 Related State of the Art 194
7.2.1 Presentation of the authentication schemes 194
7.2.2 Secure message propagation 196
7.2.3 Secure algorithms that deliver the computational load on the network 196
7.2.4 Blockchain Mechanisms in IoV Privacy Maintenance 197
7.3 Analysis of the cryptographic protocols used 198
7.3.1 Advanced Encryption Standard (AES) Cryptographic protocol 198
7.3.2 Elliptic Curve Cryptography 199
7.3.3 Hyperelliptic Curve Cryptography for genus ≥ 2 201
7.4 Which was the followed solution 203
7.4.1 Software used 204

7.4.2 Realization of the cryptographic algorithms 204
7.5 Realization of the proposed solution 213
7.5.1 Cryptographic methods 213
7.5.2 VANET communication realization 216
7.5.3 How the energy model was realized 217
7.6 Assessment of the experiments 217
7.6.1 Assessment framework 218
7.6.2 Duration of the various cryptographic operations 218
7.6.3 What size consume the exchanged messages 220
7.6.4 Energy consumption 222
7.7 An innovative device for encrypting messages through Zigbee module via the use of AES and ECC 224
7.7.1 The message encryption/decryption device 224
7.8 Conclusions and future work 230
Part 3: Discussion and Future Plans 233
List of author's publications 239
Patents 241
References 242

List of Figures

Figure 1 The 2 main basic devices used for data-logging sensed parameters form the basil pot. On the left, the Raspberry Pi and on the right the Arduino MEGA 2560 R3 with the various sensors. 54
Figure 2 View of the Arduino-based measuring device that was used to data-log Current, Voltage and Power consumption of each connected load on the USB output..... 56
Figure 3 View of the infrastructure for sensing the parameters of the basil pot (temperature, humidity, UV radiance) and the measuring device, which measures and data-logs Voltage, Current and Power consumption. 59
Figure 4 Relative Humidity (%) that was measured in the experiment. It is easy seen that humidity stays low in the day and rises in the night. 60
Figure 5 Temperature (^{o}C) that was measured in the experiment. It is risen in the day and decreases in the night. 60
Figure 6 UV radiation that was measured in the experiment. It makes sense to have non-zero values during the day, where there is plenty of light and non-zero values at night.. 60
Figure 7 Soil moisture indirect calculation via the raw data coming from the A/D converter of the Arduino during the experiment. The higher the A/D converter's value displayed in this figure the lower the actual soil moisture. When irrigation takes place "negative spikes" are observed, whereas when dry soil exists, the values are increasing.60
Figure 8 The UML diagram of the DSS (Decision Support System) proposed in the current chapter. 61
Figure 9 Indirect soil moisture from A/D converter raw values of the Arduino filtered with Savitzky-Golay filter in order to have an overall clearer picture of where the actual thresholds start and at which point, they end. 62
Figure 10 RNN basic model..... 63
Figure 11 RNN expanded model. 64
Figure 12 This image depicts the expanded single node of the RNN. 64
Figure 13 This image depicts the LSTMs cells. As it is obvious, each LSTM cell contains four layers that interact. 65
Figure 14 Representation of the LSTM memory block, which consists of one cell with 3 gated layers [48]. 65
Figure 15 UV radiance data-logging with 1 minute frequency and 20.000 values in total. The actual (initial) dataset is colored with red, predictions on trained (known) data are colored with blue color and predictions on unseen (unknown) data are colored with green. 67
Figure 16 Indirect soil moisture data-logging with 1 minute frequency and 20.000 values in total. These are raw values coming from the A/D converter of the Arduino. The actual (initial) dataset is colored with red, predictions on trained (known) data are colored with blue color and predictions on unseen (unknown) data are colored with green. 67
Figure 17 Relative Humidity data-logging with 1 minute frequency and 20.000 values in total. 67
Figure 18 Temperature data-logging with 1 minute frequency and 20.000 values in total. 67

Figure 19 Relative Humidity Loss function operating for 400 epochs training, both for validation (with orange color) and test (with blue color) values. 69
Figure 20 Soil Moisture Loss function operating for 400 epochs training, both for validation (with orange color) and test (with blue color) values. 70
Figure 21 Temperature function operating for 400 epochs training, both for validation (with orange color) and test (with blue color) values. 70
Figure 22 Power consumption of the completed circuit, measured in mWatts, with 1 hour sampling frequency. 71
Figure 23 Power consumption of the completed circuit, measured in mWatts, with 1 minute sampling frequency. 71
Figure 24 In the picture someone can see how is a signal (message) that is sent from the Arduino (existing in the basil pot) to the Raspberry Pi, via Xbee Zigbee is viewed under the oscilloscope. The current needed from the Xbee Zigbee module is around 40 mA, according the manufacturer. The whole device operates at 5 Volts, which results in a power consumption of P = V * I = 5 * 40 mA = 200 mW on peak of the signal. 72
Figure 25 a) How the 2D-convolution looks like [62]. b) How Convolution with stride s = 2 seems [62]. c) A 3x3 max pooling with stride (step) s = 2 [62]. d) How a Fully Connected level looks like [62]. 87
Figure 26 Scheme of the AlexNet neural network [62]. 89
Figure 27 CPU usage using the Pillow library. 93
Figure 28 CPU usage of each SBC, demonstrated in percentage. 94
Figure 29 Accuracy for each class. 95
Figure 30 Inference duration for every class via the use of Google Colab. 96
Figure 31 Used memory with Pillow, depicted in percentage. 96
Figure 32 Memory usage depicted in MBytes when Pillow is chosen to operate. 97
Figure 33 Memory usage of each SBC, depicted in percentage in contrast to its total available memory. 97
Figure 34 Internal temperature of the SBCs when Pillow is used. 98
Figure 35 Memory usage of SBC, depicted in percentage in contrast to its total available memory. 98
Figure 36 The current that each device needs when using Pillow**.** 99
Figure 37 Average current consumption of each SBC, presented in milli-Amperes. 100
Figure 38 CPU usage, with ImageDataGenerator and batch_size = 2. 100
Figure 39 CPU usage, with ImageDataGenerator and batch_size = 4. 101
Figure 40 CPU usage, with ImageDataGenerator and batch_size = 8. 101
Figure 41 CPU usage, with ImageDataGenerator and batch_size = 16. 102
Figure 42 Memory usage in percentage, with ImageDataGenerator and batch_size = 2. 102
Figure 43 Memory usage in percentage, with ImageDataGenerator and batch_size = 4. 103
Figure 44 Memory usage in percentage, with ImageDataGenerator and batch_size = 8. 103
Figure 45 Memory usage in percentage, with ImageDataGenerator and batch_size = 16. 103
Figure 46 Memory usage in MBytes, with ImageDataGenerator and batch_size = 2. .. 104
Figure 47 Memory usage in MBytes, with ImageDataGenerator and batch_size = 4. .. 104

Figure 48 Memory usage in MBytes, with ImageDataGenerator and batch_size = 8... 105
Figure 49 Memory usage in MBytes, with ImageDataGenerator and batch_size = 16. 105
Figure 50 Temperature of both devices with ImageDataGenerator and batch size = 2. 106
Figure 51 Temperature of both devices with ImageDataGenerator and batch size = 4. 106
Figure 52 Temperature of both devices with ImageDataGenerator and batch size = 8. 107
Figure 53 Temperature of both devices with ImageDataGenerator and batch size = 16. 107
Figure 54 Current draw in mA of both devices with ImageDataGenerator and batch size = 2. 108
Figure 55 Current draw in mA of both devices with ImageDataGenerator and batch size = 4. 108
Figure 56 Current draw in mA of both devices with ImageDataGenerator and batch size = 8. 108
Figure 57 Current draw in mA of both devices with ImageDataGenerator and batch size = 16. 109
Figure 58 The CPU usage of Raspberry Pi 3B+. 109
Figure 59 Percentage of RAM memory usage while using Raspberry Pi 3B+..... 110
Figure 60 SIze (in MBytes) of RAM memory usage while using Raspberry Pi 3B+.... 110
Figure 61 SIze (in MBytes) of Swap memory usage while using Raspberry Pi 3B+.... 111
Figure 62 Temperature behaviour of Raspberry Pi 3B+. 111
Figure 63 Raspberry Pi 4B results, concerning CPU usage. 112
Figure 64 Raspberry Pi 4B results, concerning RAM memory usage, depicted in percentage. 112
Figure 65 Raspberry Pi 4B results, concerning RAM memory usage, depicted in MBytes. 113
Figure 66 Raspberry Pi 4B temperature results. 113
Figure 67 Completion time of every SBC on the inference part. 114
Figure 68 Giga (10^9) floating point operations or Giga operations per second per 1 milli-Watt power consumption for each SBC. 115
Figure 69 Graphing example of a linear regression equation. On x-axis are the VI (Vegetation Index) values, here are the NDVI, and on y-axis are the salinity values. ... 121
Figure 70 Application for evaluating soil salinity, where the user needs to choose either RGB or Reflectance UAV images. 123
Figure 71 Assessed salinity range for the selected BGR UAV image. As it is obvious the Int1 Vegetation Index was used. 124
Figure 72 Salinity histogram for year 2021. 126
Figure 73 Salinity histogram for year 2022. 126
Figure 74 Linear Regression mapping to the related salinity values, for all 9 Vegetation Indices. 130
Figure 75 Comparison of predicted and observed salinity values extracted from **Table 12** above. 131
Figure 76 Comparison of predicted and observed water level values extracted from **Table 13** above. 132
Figure 77 Comparison of predicted and observed water temperature values extracted from **Table 14** above. 133

Figure 78 Comparison of predicted and observed water content values extracted from **Table 15** above. 134
Figure 79 Measured electrical conductivity and related coordinates while pointing the area with the mouse pointer. 139
Figure 80 Temperature forecast via the use of an RN-LSTM model. 140
Figure 81 The SSD ResNet50 V1 FPN 640x640. 142
Figure 82 The object detection output images. As it seen there are rectangle showing the related soil salinity with the respected accuracy. 143
Figure 83 The images we used as input to the object detection model. Those were the initial images before the modifications in order to keep the area we need to work on. . 144
Figure 84 Classification Loss vs. Steps. 145
Figure 85 Regularization Loss vs. Steps. 146
Figure 86 Localization loss vs. Steps. 146
Figure 87 Total loss vs. Steps. 146
Figure 88 a) Drilled technique in a tapped Pinus merkussi, **b)** Quarre technique on the same tree [86]. 150
Figure 89 The Arduino microcontroller, which is responsible for supervising all the operations from/to various sensors. 152
Figure 90 The Adafruit FONA (GSM/GPRS) module. 153
Figure 91 The Xbee Zigbee S2 2mWatt wireless module. 154
Figure 92 The external sensor for sensing temperature and air humidity in the environment. 154
Figure 93 The LCD 16x2 screen, which is in charge to depict information to the user, via triggering the external push button. 155
Figure 94 The DHT22 air humidity/temperature sensor which is used in order to have a sense of the conditions inside the device. 155
Figure 95 The Load Sensor Amplifier, which senses the resin weight in the bucket via the load sensors and sends the data to the Arduino microcontroller. 156
Figure 96 The circuit connection of the 4 load cells and the respected connections to the HX711. As it is obvious the circuit consists a wheatstone bridge. 156
Figure 97 The rain sensor enclosed into a rain-proof box. The rain sensor incorporates a 1-meter 2-wire cable, so that it can placed in the environment far from the central device. When water falls in the sensor, the Arduino controller instantly informs the end user.. 157
Figure 98 Logic table of control pins and the related output (selected device). 158
Figure 99 The TCA9548A I2C Multiplexer. 158
Figure 100 Two relay module, in charge to open/close the LCD screen and/or the fan. 159
Figure 101 The fan responsible to cool the whole circuit and the battery when there is increased temperature inside the box. 159
Figure 102 Fan shield. 160
Figure 103 Push-button, that the user uses when they want to get information in the LCD screen instantly. 160
Figure 104 ON/OFF switch to start/close the device operation. 161
Figure 105 GSM antenna of the GSM/GPRS module. 161
Figure 106 The solar power manager interface[47]. 162
Figure 107 Solar power manager with coolers attached to every side. 162

Figure 108 Fuse holder with a 600 mA fast fuse inside in order to protect the whole circuit from short circuits. 163
Figure 109 The lower metallic disc with the four load sensors attached to it. 163
Figure 110 The solar panel used in order to supply with power the resin collection device. 164
Figure 111 Overall image of the pine resin collection device with the main device, the solar pane, the rain sensor and the bucket resin collection device. 165
Figure 112 This image depicts the inside of the resin collector device, where all the elements and parts are seen. At the time of the screenshot the parts were not assembled in order to be clearly visible. 165
Figure 113 The AS operation as it depicted in a flow diagram. 172
Figure 114 The header message used in the simulation. 180
Figure 115 Time representation for the key generation, measured in ms. 181
Figure 116 Time for certificate generation in ms. 181
Figure 117 Time for the encryption phase, measured in ms. 182
Figure 118 Time for the decryption phase, measured in ms. 182
Figure 119 Time for signature generation, measured in ms. 183
Figure 120 Time for signature verification, measured in ms. 183
Figure 121 Size of messages, in bytes, during the negotiation phase. 185
Figure 122 Size of messages, in bytes, during the pseudonym exchange period. 185
Figure 123 Size of messages, in bytes, during the phase of new pseudonyms enabling. 186
Figure 124 Consumed energy, measured in Joules, during the phase of exchanging pseudonyms 186
Figure 125 Total entropy of nodes (vehicles) throughout the time, measured in seconds. 187
Figure 126 Reverse points of the Elliptic Curve [163]. 200
Figure 127 Rule of the Group for the Elliptic Curve points [163]. 200
Figure 128 This is an example a g = 2 Hyperelliptic Curve, with $y^2 = f(x)$[163]. 202
Figure 129 Generation technique linked to AES algorithm 213
Figure 130 Encryption of an AES message. 214
Figure 131 Decryption of an AES message. 214
Figure 132 Validation of the HECC genus 3 Curve. 215
Figure 133 Realization of divisor's Curve genus 3 215
Figure 134 Times for various cryptographic realizations in ms. 219
Figure 135 Size of exchanged messages when using the ns-3 simulation software. The sizes are depicted in Bytes. 221
Figure 136 Consumption of energy in very small-time realization. 223
Figure 137 Energy consumption of the various operations, given in Joules. 223
Figure 138 The encryption/decryption device with the related connections to the rest electronics. 225
Figure 139 The experiment for 2 users. Each user can type the message which is sent encrypted to the other user wirelessly via the Zigbee wirelessly. The user that receives the message can read the decrypted message in their TFT screen. 226
Figure 140 The interface of the device's TFT screen, when the user gets a decrypted message, when AES symmetric scheme is used. 227

Figure 141 The interface of the device's TFT screen, when the user gets a decrypted message, when modified ECC asymmetric scheme is used. 227
Figure 142 This figure demonstrates a comparison between the AES or modified-ECC when encrypting and decrypting the same message, when using Arduino MEGA 2560 R3 microcontroller. 228
Figure 143 The electrical footprint of the plaintext while being sent from the USB keyboard to the Arduino MEGA 2560 R3 module. Then it undergoes encryption via either the use of AES or the modified-ECC algorithm. 228
Figure 144 The footprint of the AES-encrypted message before be fed to the Zigbee module and then be transmitted wirelessly. 229
Figure 145 The time durations for encryption and decryption for the message: "Hello George!" realized in Arduino GIGA R1, using either AES scheme or modified-ECC.. 230
Figure 146 The patented electronic mailbox. On the left side someone can see the processing unit, whereas on the right side can see the mailbox which contains inside various sensors. 235
Figure 147 How the central node communicates with the rest nodes via Zigbee and the end-user via GSM/GPRS/4G/5G. 236

List of Tables

Table 1 In the table someone can view the different elements used on Figure 1 and their connection between them. 55
Table 2 In the table someone can view the different elements used on Figure 2 and their connection between them. 57
Table 3 Metrics related to performance for sensors reading evaluation targeting training part. 68
Table 4 Metrics related to performance for sensors reading evaluation targeting testing part. 68
Table 5 View of the various Schemes, their connected sensors, their micro-controllers or microprocessors, and their different power consumption (in mWatts). 73
Table 6 Comparison of the different SBCs used in the experiments 85
Table 7 GFLOPs stands for Giga (10^9) floating point operations per second and is implemented to Raspberry Pi 3B+, Raspberry Pi 4B, and NVIDIA Jetson Nano. Whereas GOPs stands for Giga (10^9) operations per second and is implemented to Google Coral Dev TPU. 114
Table 8 Linear Regression X (NDVI), Y (salinity) values. 121
Table 9 Overall statistics for both ground sensors' data in year 2021. 125
Table 10 T-test related to salinity for years 2021, 2022 for both the two ground sensors. 125
Table 11 Mapping between Vegetation Indices and the related salinity values. 129
Table 12 All the data related to regression of salinity with the different Vegetation Indices. 131
Table 13 All the data related to regression of water level with the various channels of the image. 132
Table 14 All the data related to regression of water temperature with the various channels of the image. 133
Table 15 All the data related to regression of water content with the various channels of the image. 134
Table 16 The categorization of the UAV images, used as training data or testing data, as referenced to the "FOLDER" column. Also, it is observable the salinity as it is rounded. 141
Table 17 Parameters of the training. 145
Table 18 Comparison of the different key sizes of RSA and ECC, having in common the same security level [127]. 173
Table 19 The current table shows different key sizes of RSA and ECC schemes, for the same security level with the related ratio of their key size [119]. 173
Table 20 The current table compares ECC, RSA and NTRU for the same security level [123]. 173
Table 21 Symmetric and asymmetric schemes comparison, for different security levels [128]. 174
Table 22 Comparison of PQC algorithms against classical cryptographic schemes in order to give an estimation of difficulty to be penetrated [128]. 175

Table 23 Different proposed protocols are depicted based on different research papers targeting Location Privacy Protections in IoV environments [132][133][134][135][136][137][138][139][140]..... 188

List of Abbreviations

AC	Alternating Current
ADAM	Adaptive Moment Estimation
AES	Advanced Encryption Standard
AI	Artificial Intelligence
AIL	Ambient Intelligence Laboratory
ALI	Anonymous Lightweight Inter-vehicle
ALRS	Anonymous and Linkable Ring Signcryption
ANN	Artificial Neural Network
ANOVA	ANalysis Of Variance
AODV	Ad hoc On-Demand Distance Vector
API	Application Programming Interface
APP	Application
ARIMA	Autoregressive Integrated Moving Average
ARM	Acorn RISC Machine
ASIC	Application Specific Integration Circuit
BC	Blue Color
BGR	Blue Green Red
BLE	Bluetooth Low Energy
CC	Correlation Coefficient
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSV	Comma Separated Value
CUDA	Compute Unified Device Architecture
CWM	Cluster-Weighted Modelling
DC	Direct Current
DH	Diffie-Hellman
DID	Dummy Identification
DIV	Division
DPSZ	Dynamic Pseudonym Swap Zone
DSA	Digital Signature Algorithm
DSS	Decision Support Systems
DW	Durbin Watson
EAAP	Efficient Anonymous Authentication
EC	Electrical Conductivity
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
EEPROM	Electrically Erasable Programmable Read-Only Memory
EMI	Electro Magnetic Interference
eMMC	embedded MultiMedia Card

EVI	Enhanced Vegetation Index
FPN	Feature Pyramid Network
GB	Giga Byte
GC	Green Color
GDAL	Geospatial Data Abstraction Library
GL	Group Leader
GND	Ground pin
GPIO	General Purpose Input Output
GPRS	General Packet Radio Service
GPU	Graphics Processing Unit
GRRN	Green Red RedEdge NearInfrared
GSM	Global System for Mobile Communications
HAT	Hardware Attached on Top
HD	High Definition
HDD	Hard Disk Drive
HDMI	High-Definition Multimedia Interface
HECC	Hyper Elliptic Curve Cryptography
I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
ITS	Intelligent Transportation System
JB	Jarque-Bera
JSON	JavaScript Object Notation
KB	Kilo Byte
LAN	Local Area Network
LCD	Liquid Crystals Display
LDR	Light Dependent Resistors
LORA	Long Range
LPDDR4	Low-Power Double Data Rate - 4
LR	Logistic Regression
LSTM	Long Short Term Network
LTE	Long Term Evolution
MAE	Mean Absolute Error
MIMO	Multiple-Input and Multiple-Output
ML	Montgomery Ladder
MLR	Multiple Linear Regression
MSAVI2	Modified Soil Adjusted Vegetation Index 2
MSE	Mean Square Error
MSI	Multispectral Image
MUX	MUltipleXer
NAF	Non-adjacent Form
NDVI	Normalized Difference Vegetation Index

NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NTL	Number Theory Library
NTRU	Number Theory Research Unit
NTT	Number Theoretic Transform
OLI	Operational Land Imager
OS	Operating System
OSI	Open Systems Interconnection
OUT	Output
PC	Personal Computer
PCA	Principal Component Analysis
PLS	Partial Least Squares
PPDAS	Privacy-Preserving Dual Authentication and Key Agreement Scheme
PWM	Pulse Width Modulation
QGIS	Q Geographic Information System
R2	Squared
RAE	Relative Absolute Error
RAM	Random Access Memory
RC	Red Color
R-CNN	Region-based Convolutional Neural Networks
ReLU	Rectified Linear Unit
RENDVI	Red Edge Normalized Difference Vegetation Index
RF	Radio Frequency
RF	Random Forest
RH	Relative Humidity
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RRSE	Root Relative Absolute Error
RSA	Rivest-Shamir-Adleman
RSE	Root Square Error
RSU	Road Side Unit
RVI	Ratio Vegetation Index
SAM	Square and Multiply
SARIMA	Seasonal Auto-Regressive Integrated Moving Average
SAS	Salt-Affected Soils
SAVI	Soil-Adjusted Vegetation Index
SBC	Single Board Computer
SCL	Serial Clock
SD	Secure Digital
SDA	Serial Data
SDK	Software Development Kit
SDRAM	Synchronous Dynamic Random-Access Memory
SHA	Secure Hash Algorithms

SI	Soil Index
SLR	Simple Linear Regression
SMA	SubMiniature version A
SMS	Short Message/Messaging Service
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SSD	Single Shot Detector
SUMO	Simulation of Urban Mobility
SVM	Support Vector Machine
TA	Trusted Authority
TDMA	Time Division Multiple Access
TFT	Thin-Film-Transistor
TOPS	Trillion Operations Per Second
TPU	Tensor Processing Unit
UART	Universal Asynchronous Receiver-Transmitter
UAV	Unmanned Aerial Vehicle
UML	Unified Modelling Language
URL	Uniform Resource Locator
USB	Universal Serial Bus
UV	Ultra Violet
VANET	Vehicular Ad hoc NETwork
VCC	Voltage positive pin
VI	Vegetation Index
VM	Virtual Machine
WAVE	Wireless Access Vehicular Environment
WiFi	Wireless Fidelity
WLAN	Wireless LAN
XML	Extensible Markup Language
XOR	eXclusive OR
YOLO	You Only Look Once (neural network)
YOLOv3	You Only Look Once - 3

Preface

Acknowledgements

I would like to thank my supervisor, Associate Professor Ioanna Roussaki, who believed in me and gave me the chance to follow the current PhD program and perform research in an area that I always loved, the Internet of Things. Without her continuous support in the field, I would not have achieved this. She also helped me take part in scholarships and European projects, in order to realize my PhD research.

I would like to thank the three-member committee, for their support and the chance they gave me to follow the current PhD program.

Special thanks to the people of the Ambient Intelligence Laboratory (AIL) for their support when submitting journals and papers and their general cooperation in the various projects related to the current PhD.

Many thanks to my friends that supported me throughout the years.

My family was the cornerstone of my Ph.D. journey, as has always been in my life. My wonderful parents, Christos and Loukia, were always there to embrace me with love, to advise and support me, as was my beloved brother, Dimitrios, who cares for me unconditionally.

Structure

The thesis is structured as follows:
“Part 1: Knowledge Extraction in Internet of Things” starts with one of the main prototypes I built for the agriculture-related part of my research, which is presented in Chapter 1. It is an Internet of Things (IoT) device, incorporating Arduino microcontroller and various sensors, targeting sensing various parameters related to agriculture. The Machine Learning model that was used in order to predict future values of the sensed parameters and help the users take significant decisions about their farms is also presented here.
In Chapter 2, there are many experiments on IoT Single Board Computer (SBC) devices with different architectural processing units (CPU, GPU, TPU). Machine Learning code targeting agriculture was executed on these devices and various metrics were gathered, in order to evaluate them.

In Chapter 3, we analyze an application using linear and multiple regression in order to estimate the Rice fields’ salinity with a number of IoT devices placed in the ground, but mostly based on UAV images.

Another implementation is presented in Chapter 4, which predicts future environmental parameters regarding Maize fields and soil salinity in Rice fields, using ground IoT sensors and Machine Learning models, such as Convolutional Neural Network (CNN) and Recurrent Neural Network - Long Short-Term Memory (RNN-LSTM).

In Chapter 5, there is a presentation of a new IoT device for monitoring pine resin (or rubber) collection via various sensors (weight, temperature, humidity, rain) and informing the end user via a Global System for Mobile Communications/General Packet Radio Service (GSM/GPRS) or Zigbee module.

“Part 2: Security in Internet of Things” focuses on another aspect of IoT. In Chapter 6, we present the IoT rationale in the Internet of Vehicles (IoV). We analyze the IoV privacy in vehicles via the use of asymmetric cryptographic protocols, such as RSA, NTRU, ECC, and a symmetric cryptographic protocol, such as the AES. We also implement a specific modern protocol and gather various metrics related to IoV privacy.

In Chapter 7, we implement a new algorithm for protecting privacy in IoV, using three asymmetric cryptographic protocols, such as ECC, HECC genus 2 and HECC genus 3, and the symmetric AES. We have conducted many experiments concerning Road Side Unit (RSU), Group Leader (GL) and vehicles, and gathered many useful metrics.

Finally, in “Part 3: Discussion and Future Plans”, we present our conclusions, as well as our vision and the plans for the next steps in our research.

Part 1: Knowledge Extraction in Internet of Things

Chapter 1: Internet of Things technology in precision agriculture

1.1 Introduction

Agriculture maintains an essential role in the society and the community in a worldwide level. There are many difficulties to bypass in that area, for example security of food, correct uses of the various resources that exist in nature, the variations of the climate, the more demands of food, footprint control, biodiversity wastage and many others. Internet of Things is a technology that can help to give a solution to many of the aforementioned problems, by enabling precision agriculture, as part of Agriculture 4.0, and bringing best results for each occasion. One essential challenge is to mitigate the water usage in agriculture.

IoT realizes the connection between many devices, in order to interact by exchanging data and taking advantage of Cloud services. We can gather data from various IoT places and process it to support decisions related to optimization [1]. IoT technologies, nowadays have entered the commercial sector and significant effort is being spent to make them more user-friendly for people who do not have extensive background knowledge. IoT has shown a tremendous potential for enhancing people's life and improving many operations in various domains, such as health, education, manufacturing, and agriculture [2]. Agriculture is a great example, as the spreading of IoT technology has actually guided us towards Agriculture 4.0. One of the essential difficulties that exist today and is related to the food system in a worldwide level is the need to raise food production by half more (50%) until year 2050, in relation with 2010 [3]. The latter is critical to occur in order to provide food to a foreseen population of about 10 billion people, while in parallel there are rising stressed and finite resources (in nature) and needing to adjust to fast changing climate conditions. However, today's environmental conditions do not help to the needed increase in crop production with the traditional approach in agricultural handling [3]. All the previous described technologies can help in solving this problem, and bring increase in yields while also protecting the precious resources. At this point, precision agriculture and intelligent farming can make a great difference. The acceptance of accurate irrigation assists farmers in order to use water only for the crops that really need it. The result is the saving of water resources.

It is known [4] that the United States have a lot of problems with water availability. In California, for example, at the year 2019, in order to irrigate 1,530,000 almond acres, the farmers used 195.26 billion gallons per year. A similar situation seems to be troubling China [5]. The irrigation system provides support to the agriculture and socio-economic sectors. To further extend it, it maintains a balance in the environment which is in charge for the surviving of the oasis ecosystem. The recent years, due to people's work and because of the climatological changes, there is a change of the water that exists in arid areas (oasis). The consequence of this, is that the underground water reservoirs, the water needed for farm activity and the corresponding water salinity, are all affected. According to the World Bank, around 70% of the water is necessary for the agriculture, while factories and the rest industry need around 20%. The other 10% of water is used internationally for household works. By the year 2050, people living on Earth will reach the amazing number of 10 billion. It is obvious that the needs for water and food will increase [6].

1.2 Related State of the Art

There are plenty of initiatives that research the viewpoints of smart farming, related to the benefits of the Internet of Things technology, as well as Machine Learning algorithms, control of energy that IoT devices draw, and many others. A lot of research work effort has been focusing on the IoT field [7] [8] [9] and on Artificial Intelligence [10] [11], aiming on farm irrigation. The most significant efforts related to the state-of-the-art research on IoT and/or Machine Learning algorithms with the view of reducing water in irrigation of farms as well as decrease of energy consumption without affecting crops are presented here.

1.2.1 Smart irrigation approaches focusing on IoT and ML

In [12], the authors refer to a smart irrigation device that is controlled via IoT technology by the user. It can update and inform the user on the values of soil moisture, temperature on crops and pH using special sensors. The sensors communicate with Arduino in order to send the sensed values. The proposed devices help to decrease the water logging in the farm fields. When the temperature increases, the idea is that the system uses drip irrigation so that it can chill the crops. The pH sensor informs the user so that it can take measures if the soil is acid, which affects crop not to grow. Finally, the device updates the user for critical situations of the crops.

In [13], the authors describe a system that takes good advantage of soil moisture and the devices that control water irrigation, and when the soil wetness exceeds a certain value, it changes the time of irrigation. They use an Arduino microcontroller programmed by Arduino Integrated Development Environment (IDE). When the values of the soil moisture sensors fall below certain values the related water pumps start operating in order to irrigate. However, when the moisture values are over specific values, the irrigation enters a hold on.

In [13] the researchers propose a device that informs the user in real-time on various irrigation aspects. It aims to reduce the cost and use optimally the resources. For instance, it data-logs temperature and soil humidity, by setting different ranges for the soil and crop types. The irrigation system switches on or off according to the related thresholds that are exceeded. The device switches the related water pumps on or off according to the values it senses. The sensors, such as soil moisture, pH sensor and temperature sensor are connected to an Arduino MEGA 2560 micro-controller. Moreover, the researchers have built an Android application, in order to handle the water pump either via GSM cellular network or via Bluetooth of the Android phone.

In [14] the authors use wireless receivers/transceivers, in order to realize smart communication while limiting power consumption. Their constructure uses: a 50-watt solar panel (consisted of a 5Volt Li-ion battery), an ESP8266 microcontroller, sensors for measuring humidity and temperature and an anemometer. The data processing and the related analytics of the machine are transmitted via Cloud infrastructure. The latter contains historical data in order to make good use of them and feed decision support on when to irrigate. The authors have also connected an LCD module with the target to display more user-friendly parameters to the operator.

In another endeavour [15] they constructed a machine consisted of a sensor for measuring soil moisture, a temperature sensor, a humidity sensor, and a pH sensor. They have implemented an

application that can be used by the operators (farmers) and the consumers. The latter can order vegetables or other goods via that application. The device is powered by a solar panel. An Arduino microcontroller gathers the data from the sensors and feed the Cloud while filling the data measurements with timestamps. As far as the Machine Learning part is concerned, the authors followed the Google Inception v2 model, and used 90.000 images gathered from 38 classes, 20% of them for testing and 80% for training. The accuracy reached 96.8% on the training part and 96.4% on validation part. The purpose of all these is that users can monitor the health of the farm fields, get informed about possible deceases and buy from them when they feel it is the right time.

In [16], the authors have built a system based on Arduino Uno R3 with the following sensors: JXCT soil sensor, pH sensor, soil moisture, a sensor to measure soil temperature and a micronutrient sensor, an Electrical Conductivity (EC) sensor, an OC sensor, and an OM sensor. The Arduino microcontroller sends all the measurements to a Server through the help of an ESP8266 Wireless LAN (WLAN). Then, a Kafka cluster is used, in order to send the data to an ML server, so that it can train them. Extensive analysis takes place by considering specific thresholds on the data.

In [17], they have made a device which incorporates a Raspberry Pi 3 and the famous NodeMCU Devkit. They connected a DHT-11 sensor and a Hygrometer. This device gathers measurements from the sensors and send them to a DataBase. DHT-11 is a humidity and temperature sensor. A motor, also, is used for bringing water in cases the moisture falls below certain values. The proposed device stores all the measurements in Firebase, in order to use them for future prediction. More than 700 records are included. They divided the data for training and testing. A prediction of 3 of the most appropriate crops can be made, presenting also a number of how confident is that prediction.

All the previous presented work cannot be scaled easily. The devices and algorithms can work well in small environments, but when they are deployed in large scale there are some difficulties, such as the fact that they are powered by batteries on a 24-hour operation, 7 days a week, using a solar panel. However, there are many countries especially in the North where they have 80% - 90% clouds or rain and no sun at all. Another problem is the fact that the Raspberry's CPU gets hot and a fan should be used, thus consuming more current, that is precious in a battery-powered device. Arduino do not face cooling issues with their CPUs because the mainstream boards such as UNO, MEGA 2560 have their CPUs clocked at 16MHz. Another issue is that sensors because they are very cheap and use most of them for lab experiments cannot be used for real cases: their (Mean Time Between Failures) MTBF is really low. GPRS also draw current, when they meet "spikes", they can draw 1A or 2A currents. LoRa (Long Range), Xbee are more suitable for communicating between IoT devices, but WIFI not. WIFI (Wireless Fidelity) draws more current than LoRa and Xbee and covers distances up to 100 meters, unsuitable for a farm that can cover kilometers. UV sensors also help measure UV radiation, which affects the plants as it will be presented in other part of the current thesis. None of the aforementioned examples uses RNN-LSTM Machine Learning models that are becoming the number 1 solution for forecasting in fields like agriculture [18]. RNN-LSTM neural networks are based on seasonality and can "monitor" patterns that can be used in order to then forecast. The proposed solution is based on many measurements, something critical for expecting correct work of an RNN-LSTM. Many configurations were made so that the ML could work. For instance: number of epochs, type of optimizer, dropout value, learning rate, etc.

1.2.2 Current research towards energy consumption control

In [19] the researchers describe current experiments concerning energy consumption in IoT devices powered by batteries and used in the agriculture area. They communicate with the devices via 3G/GPRS modules in order to aggregate temperature, humidity and noise signals when somebody logs trees illegally. LoRa modules are used on dairy areas. The researchers also depict GPRS modem curve when the modem rises up to 200 mA, something that indicates the idea that GPRS modules consume a lot of energy in comparison to other Receiver/Transceiver RX/TX wireless modules.

In [20] the researchers have made an IoT module consuming low power for use in agriculture field. They describe how much time can power be provided by the battery, and especially mAh, using calculations and various measurements. They use LoRa receivers/transmitters. As they claim, although the current consumption is not high during the sleep state of the device, it has consequences in the battery of the life. Moreover, they propose that if someone needs to have a battery that lasts many hours or days in his device, he should place a battery that self-discharges really slow in the LoRa module. The authors, also measured the LoRa module and they resulted in the following: the module (CMWX1ZZABZ) has a consumption of about 47 mA of SF7 mode and 128 mA on SF12 during the transmission of data and 21.5 mA during the reception of the data. A calculated mean value of the current would be the following: (47 + 21.5)/2 = 34.25 mA during SF7 mode and (128 + 21.5)/2 = 74.75 mA during SF12 mode.

In [21] the researchers describe a device that aggregates data from many sensors sensing the following: temperature in the weather, soil moisture, how acid the ground is, etc. Then, further processing and analysis takes place. The authors present a table demonstrating the various IoT wireless devices and their current consumption for every sensor. For instance, the Xbee consumes 100 mWatts, LoRa consumes 440 mWatts, (Narrowband Internet of Things) NB-IoT consumes 550 mWatts and 5G consumes 400 mWatts. It is more than obvious that Zigbee consumes the least power among all the modules. The mainstream supply is 5 Volts, and for that number Zigbee modules draws 20 mA, LoRa draws 88 mA, NB-IoT draws 110 mA and 5G TX/RX module draws 80 mA.

In [22] they implement experiments using 2 different GSM devices suitable for use in embedded modules consisting of constrained resources. They identified through measurements what the output represents in general, without concentrate on the GSM module. Their device can measure the speed of wind, temperature, humidity, rainfall speed direction, radiation emitted by the sun and the wetness of leaves. A GL86-QUAD and a SIM900 GPRS are used so that they can send data. In their article they included graphs where it is more than obvious that about 150 mA is needed for sending SMS and about 140 mA is needed while using GPRS.

In [23] the researchers monitor the aggregated data and their related energy consumption of WSN (Wireless Sensor Networks) implemented in Farm industry. They claim about the drawbacks that WSNs meet. In their research they also present research based on IoT data processing. Different wireless devices are presented, so, for instance a BLE (Bluetooth Low Energy) module needs 10 mWatts while operating, Zigbee needs 36.9 mW, LoRa needs 100 mW, SigFox needs 122 mW, the common Bluetooth needs 215 mW, the LTE (Long-Term Evolution) is energy-hungry and needs 300 mW, GPRS even more needs 835 mW. BLE module can be a straight forward solution, however, in relation to the Zigbee, the latter covers a 100-meter area and BLE covers a 10-meter

area. What comes out for this paper is the fact that Zigbee is the less power consuming module of all the choices, adding an acceptable radio coverage.

In [24] the authors propose a machine which operates autonomously powered by solar radiation. It has many sensors connected, such as: the famous DS18B20 thermometer, the BME280 for sensing humidity and pressure, CO2 sensor, AMS CCS811 metal-oxide sensor, the FC28 for sensing soil moisture and the GL55 light sensor. The device communicates to a mobile APP (Application) in order to provide data to the user related to crops. It uses a WIFI module to send data to the API (Application Programming Interface) and it needs 5 Volt Voltage supply and an average value of 230 mA (260 mA at peak values). The average power consumption can be easily calculated as P = V * I = 5 * 230 = 1150 mW or P = 5 * 260 = 1300 mW when reaching peak values.

In [25], they propose a solution using wide-area mesh network in applications related to IoT technology, implemented in agriculture field. LoRa modules were chosen for their device, using TDMA (Time Division Multiple Access) technology for large area communication. As far as the hardware is concerned, they connected the following sensors: soil moisture sensors, weight sensors, temperature sensors, humidity sensors. A 5-Volt is used to supply the device. The depicted information, where the LoRa RT/TX module needs 12.5 mA current for reception and 72.5 mA for transmission, for a mean value of (12.5 + 72.5)/2 = 42.5 mA.

1.3 Analysis of the problem and the related proposed solution

1.3.1 What is the problem

IoT is in a very mature level in order to be implemented in the agriculture field and provide solution to many issues, such as sustainability, quality and quantity in the yield, cost effectiveness [26]. Systems of smart irrigation are being developed around IoT devices, consisting of sensors, CPUs, actuators, targeting on estimating many parameters, such as soil condition, crop, weather phenomena and provide support to related decisions made on plant irrigation.

1.3.2 Proposed solution to the problem

The solution described below has as its basis the Arduino MEGA 2560 R3 processing unit, programmed by the famous Arduino IDE (Integrated Development Environment). The module is connected with the following sensors: i) Capacitive sensors for sensing soil moisture, ii) DHT22 sensors for measuring temperature/humidity, iii) VEML6070 Ultra Violet sensor for sensing the levels of UV light in the plant.

The good thing with the capacitance soil sensors is that they cannot be corroded as occurs with the resistance soil moisture sensors. The soil moisture sensors used in the proposed device needs 5 Volt and are consisted of 3 pins, voltage positive pin (VCC), (ground) voltage negative pin (GND), output (OUT). The signal that a micro-controller can read from the soil moisture sensor is in the range 0 to 5 Volts, which is related to the intense of the moisture in the soil. The OUT signal is connected to the Arduino to one of its A/D ports. Arduino can understand 1024 different values from the sensor's output values, because Arduino's A/D converter is 10-bit = 2^{10} = 1024, so a range from 0 to 1023. More sensors were used that operate auxiliary to the primary, in order to have 2 measurements for the same condition and provide the final output.

The DHT22[3] module senses temperature and humidity in the air. It is placed in a stable place on the plant so that it can provide measurements from plant's leaves. It can operate either at 3 Volt logic or 5 Volt logic, that means both power supply and the wire communication with the rest devices. It needs 2.5 mA max, it can measure humidity in the range 0 – 100 %, with 2% to 5% accuracy, while it can measure temperatures ranging from -40 °C to 80 °C, providing +/- 0.5 °C accuracy, with 0.5 Hz sampling frequency. The sensor consists of 2 parts: a humidity capacitance sensor and a thermistor. Inside the module there is a chip for transforming Analog measurements to Digital values, so that it can send them to its output pin. The connection between the sensor and the Arduino is being settle via the digital input of the Arduino. DHT22 uses 3 pins: VCC, GND and OUT.

The VEML6070 UV[4] module works with both 3 Volts or 5 Volts power supply and logic. It exploits I2C protocol to exchange data with the Arduino. It uses the Ultra Violet spectrum to sense light. What it basically does is to output a number for each UV light level it senses. The pins used from this sensor are: VCC, GND, SDA (Serial Data), SCL (Serial Clock).

All the sensors communicate with the Arduino MEGA 2560 R3 module which uses the ATmega2560 microcontroller[5]. It has 54 digital input/output pins, 16 of them can operate as analog inputs and 4 of the 54 inputs can operate as 4 UARTs. Its CPU is clocked at 16 MHz, lower than the mainstream CPU modules, but very suitable for what is needed for the experiments. It incorporates 256KB flash, 8KB SRAM (Static Random-Access Memory) and 4KB EEPROM (Electrically Erasable Programmable Read-Only Memory) memory. Every of its input/output pins can tolerate 20 mA current draw. The communication between the Arduino and the capacitance soil sensors is configured via the pins A0 and A1 where A/Ds exist. Digital pin 2 of the Arduino is connected to the DHT22 sensor and pins 20 (SDA) and 21 (SCL), which consist the I2C protocol, are connected with the UV sensor. UART (Universal Asynchronous Receiver/Transmitter) 3 port of the Arduino is connected to the Xbee Zigbee S2 module. All the sensors and the Arduino are powered by a 5-Volt power supply and obey to 5 Volts logic.

All the data gathered from the Arduino via the sensors, are wirelessly transmitted to a Raspberry Pi 4B Single Board Computer. The modules used for this communication between the two boards, meaning the Arduino and the Raspberry, are two Xbee Zigbee S2 2mWatts Wire Antenna. Xbee works on 3.3 Volts both power and I/O in its pins. To make it work on 5 Volts, special adapters were used that include voltage level translators. For Raspberry the SparkFun Xbee Explorer USB (Universal Serial Bus) was used and for Arduino the SparkFun Xbee Explorer Regulated was used.

DIGI manufacturer produces the Xbee Zigbee. It contains improvements on power output and as well as the protocol in the Pro Series 2 that was used in the experiments. It needs 3.3 Volts and 41 mA. It emits 2 mW radiation, when used, operating at 250 kbps bitrate, which covers the requirements of the implementation. It can reach 120 meters coverage at LOS (Line of Sight), using only a small fixed wire antenna without complex installations. It enables input pins and 8 digital input/output pins. One great characteristic is its ability to encrypt and decrypt data using 128-bit encryption, configuration over-the-air as well as AT/API commands[6].

[3] https://learn.adafruit.com/dht?view=all
[4] https://learn.adafruit.com/adafruit-veml6070-uv-light-sensor-breakout?view=all
[5] https://store.arduino.cc/arduino-mega-2560-rev3
[6] https://www.sparkfun.com/products/retired/10421

Raspberry Pi 4B[7] is the most advanced SBC of the well-known Raspberry Pi family. It consists of the BCD2711 ARM (Acorn RISC Machine) CPU, which has 4-cores, 64-bit SoC clocked at 1.5 GHz with 4GB RAM. It provides access for 2 bands WiFI: at 2.4 GHz and at 5 GHz IEEE (Institute of Electrical and Electronics Engineers) 802.11b/g/n/ac, Bluetooth 5.0, BLE, Gigabit Ethernet, 2 USB 3.0 ports and 2 USB 2.0 ports. It supports the standard 40-pin General Purpose Input Output (GPIO) header. It has 2 micro-HDMI (High-Definition Multimedia Interface) ports in order to enable connection with external monitor/monitors. The OS (Operating System) (Raspbian) is executed on a flashed 128 GB microSD. As far as the power is concerned, it is powered by a 5Volt/3 Amperes power adapter via a USB-C plug. It uses a fan in order to decrease CPU's temperature when working for many hours. Also, a keyboard and a mouse are used so that someone can operate the Raspberry Pi as a Desktop computer and make the programming easier.

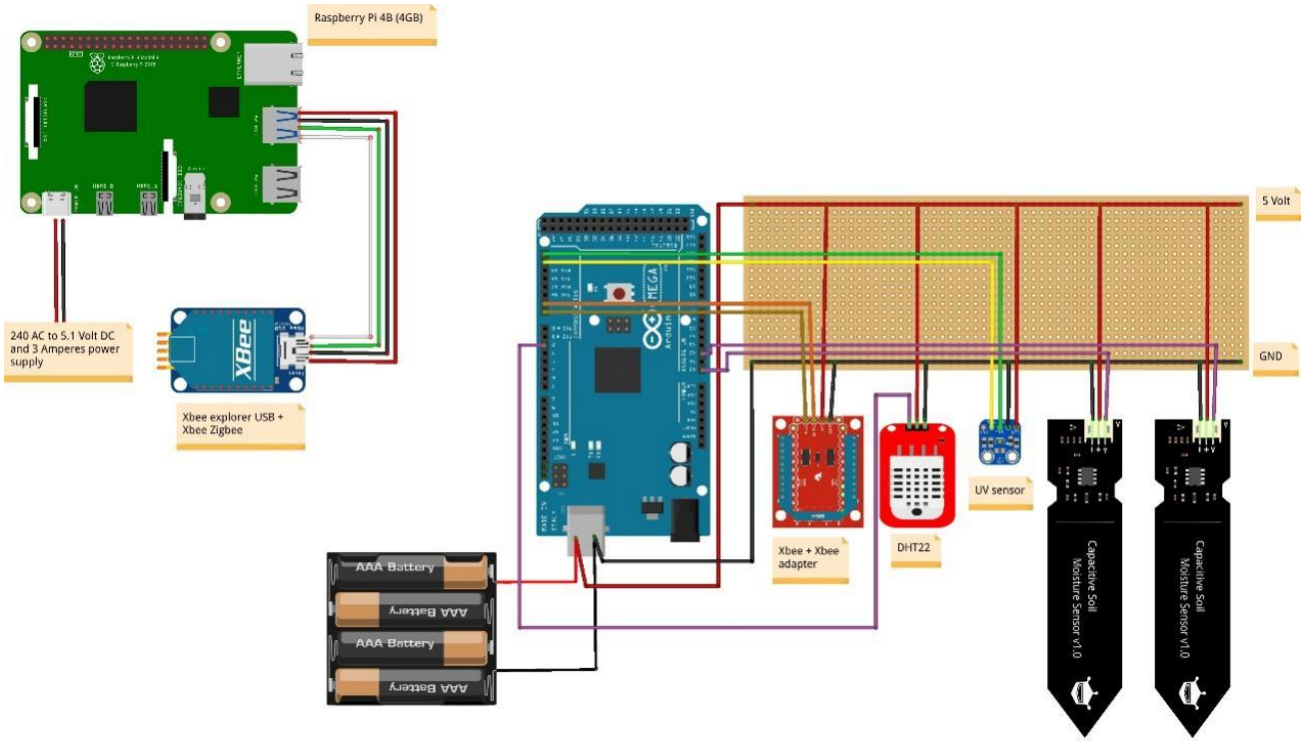


Figure 1 The 2 main basic devices used for data-logging sensed parameters form the basil pot. On the left, the Raspberry Pi and on the right the Arduino MEGA 2560 R3 with the various sensors.

The modules that were presented in the previous paragraphs are shown in **Figure 1**. The Raspberry Pi is presented at the top left side of the figure with Xbee adapter connected with it via the USB wires. It is also shown the power adapter connected to the power port of Raspberry Pi. At the right side of the figure, someone can identify the Arduino MEGA 2560 R3 with the different

[7] https://static.raspberrypi.org/files/product-briefs/200521+Raspberry+Pi+4+Product+Brief.pdf

sensors (soil moisture sensors, DHT22, UV sensor, Xbee Zigbee) connected to a proto-board and a powerbank.

Element 1	Element 2	Element 3
Raspberry Pi 4B	240Volt AC to 5.1Volt DC adapter	Xbee Zigbee module
Xbee Zigbee adapter USB	Xbee Zigbee module	
5.1 Volt powerbank	Arduino MEGA 2560 R3	
Xbee + Xbee adapter regulated	Arduino MEGA 2560 R3	
DHT22	Arduino MEGA 2560 R3	
UV sensor	Arduino MEGA 2560 R3	
Capacitance soil moisture sensor	Arduino MEGA 2560 R3	

Table 1 In the table someone can view the different elements used on Figure 1 and their connection between them.

In **Table 1** someone can see the connections between each element. As it was presented in the previous sections, the proposed solution has to do with the building of an Arduino-based system with many appropriate sensors in order to measure different conditions of farm field such as: soil moisture, temperature, air humidity and UV radiance with the aim to send all the aggregated data on the Single Board Computer and undergo further processing. What is very innovative is the existence of an RNN-LSTM model, programmed in python programming language, running on Ubuntu OS, capable of forecasting the following parameters: UV radiance, temperature, soil moisture and air humidity. So, the farmer or the person that is handling the operation can decide the volume of water which is needed in order to precise irrigate his farm so that no unwanted water is used, saving resources and money. Research to related articles [27] [28] [29] [30] has shown that ML algorithms perform outstanding in forecasting in many domains, including agriculture, showing decreased error (such as RMSE – Root Mean Square Error, MAE – Mean Absolute Error, MSE – Mean Square Error, CC – Correlation Coefficient) in relation to the rest of the ML models. Although RNN-LSTM implementation has limited investigation in smart irrigation problems, as it is more than obvious from the State-of-the-Art paragraphs, the usage in order to support DSS (Decision Support Systems) is very innovative.

1.4 Evaluation of the experiments that took place

In order to identify whether the previously described device meets the authors standards, an energy measuring device was built. The latter device aims to data log various characteristics of the Arduino-based IoT device concerning energy consumption, such as: timestamps of the measurements, current draw, voltage, power consumption. This device is capable of measuring Current (in milli Amperes), Voltage (in Volts) and Power consumption (in milli Watts). The upper limit of Current it can measure is + 3.2 Amperes at 5 Volts (maximum) by using the famous INA219 module via I2C port. So, the device consists of the following elements:

- an industrial shielded power supply that can provide 12 Volts, with maximum 6 Amperes current, 72 Watts.
- 2 DC (Direct Current) -DC Step-Down converters that can provide an output of 5 Volts at 5 Amperes max current. These modules provide power supply to the Arduino micro-controller and the various rest electronics, connected to the proto-board.

- INA 219 current/voltage/power measurement module.
- WiFi module in order to access the measurements wirelessly.
- Voltage - level translators in order to communicate with the WiFi module that operates at 3.3 Volt and not at 5 Volt (as occurs with the rest of the elements).
- Ethernet Shield, in order to access the measurements via LAN network.
- Arduino MEGA 2560 R3 as the main microcontroller.
- Real-Time-Clock (RTC DS3234) in order to store a timestamp with each measurement.
- SD card module, for storing the data that are being data logged.
- 240 Volts/50 Hz plug for supplying the whole construction.

All these connections are presented in **Figure 2**. The format of a stored measurement is as follows: 21/12/22,3:27:43,5.14,869.90,4742.00, an ordinary CSV format, where from left to right someone can observe the timestamp (date and time): 21/12/22,3:27:43,5, then is Voltage in Volts: 5.14, Current in milli Amperes: 869.90, Power consumption in milli Watts: 4742.00. The SD (Secure Digital) - card stores each measurement: however, they can be accessed in real-time via:

1. USB (through direct connection to a PC (Personal Computer)/laptop)
2. Ethernet
3. WiFi

The "brain" of measuring device is the Arduino MEGA 2560 micro-controller, which communicates with the peripheral modules. So, it communicates with INA219 via I2C protocol, with WiFi via one of its UART ports, with RTC via SPI (Serial Peripheral Interface) port. Below, in **Table 2** someone can observe all the elements of the measuring device and their related connections between them.

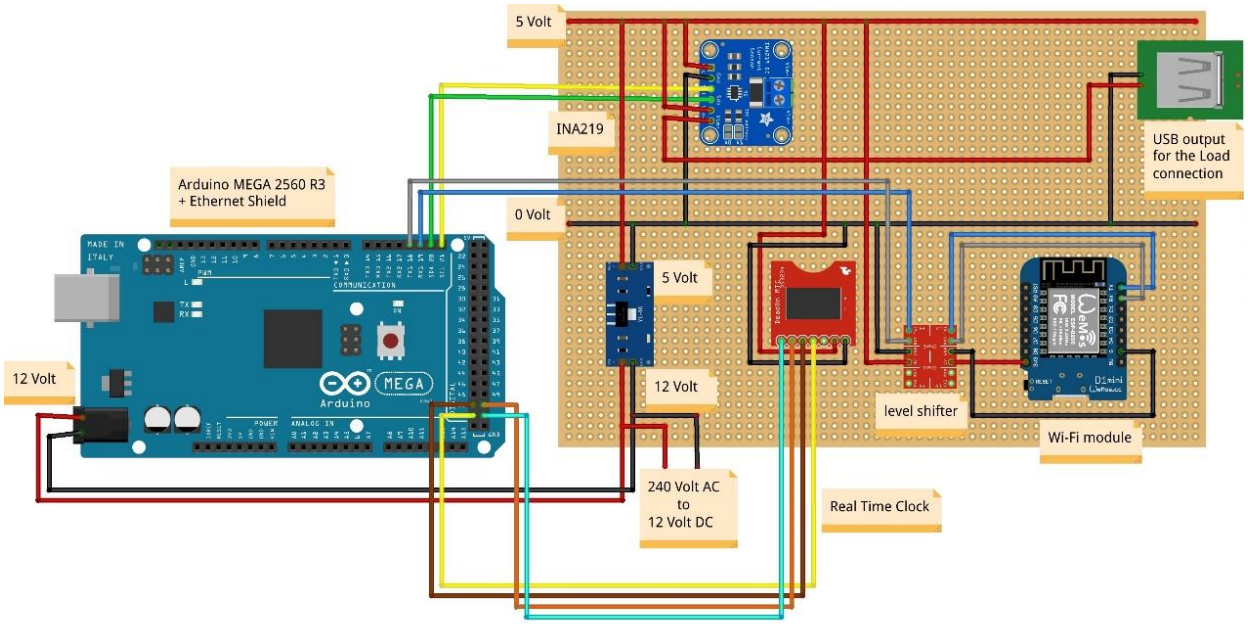


Figure 2 View of the Arduino-based measuring device that was used to data-log Current, Voltage and Power consumption of each connected load on the USB output.

Element 1	Element 2	Element 3
INA 219 current sensor	Arduino MEGA 2560 R3	
DC-DC step-down converter	240Volt AC to 12 Volt DC adapter	
Wemos D1 WiFi module	level shifter	Arduino MEGA 2560 R3
USB output	0-5 Volt in circuit supply	
level shifter	Arduino MEGA 2560 R3	Wemos D1 WiFi module
RTC DS3234	Arduino MEGA 2560 R3	
Ethernet shield	Arduino MEGA 2560 R3	

Table 2 In the table someone can view the different elements used on Figure 2 and their connection between them.

Analysis of the effects of the environment towards plants in general

As far as the wavelength is concerned, the UV radiation can be categorized into three areas:

- UV-A in the range 315 nm to 400 nm
- UV-B in the range 280 nm to 315 nm
- UV-C in the range 100 nm to 280 nm

In the Arduino-based device the connected UV sensor operates in the UV-A range. It is well known to the agriculture researchers that UV-C radiation is totally absorbed through the ozone layer. That does not occur in UV-A, which also does not harm the plants. However, UV-B is affected by the ozone layer and more specifically it affects its intensity. The most harmful of the above mentioned 3 ranges, is the UV-C radiation type [31]. According to research literature, UV-A radiation shows that adverse consequences are present on plants. In [32] the authors claim that Rosa hybrida and Fuchsia hybrida indicate response in more uniform way than to UV-B. It is believed that UV-A does not damage plant, however, in the research they show that UV-A causes harm in photosystem II. PAR which is better known as Photosynthetically Active Radiation, in the range between 400 nm to 700 nm, filled in with UV-A can increase carotenoids, chlorophyl, which consist the pigments and antioxidants (UV- absorbing compounds). All these have effect on the growth [33] [34] [35] [36] [37] [38]. Researchers believe that UV-A radiation comfort the damaging consequences of the UV-B. For that reason, low levels of UV-A can enhance the collection of antioxidants in plants which could improve the health of species that consume them.

According to experiments, the Genovese basil plants need more water at the end of development as well as at the growth stages and during the maturity stage [39]. At the starting stages of crops, the transpiration of basil is not high because of the shrunk area that is foliaged. However, the water loss is increased as a result of the evaporation rather than the transpiration that takes place on the plant [40]. During the initial stage (growth, development), precipitation and high temperatures were observed, exploiting water evaporation from the ground. While the plant was growing, there was the need to make biomass produce flowers and rise transpiration: all these resulted to increased need for water. The numbers gathered from measurements indicate that water consumption on growth and development levels of basil was 2.98 mm/day. As far as the dry mass is concerned the mean value was 27.2%. Authors in [39] work claim that the plant in its maturity period needed 4.87 mm/day (38.9% more than then needs in the previous period). However, there were no changes observed in the measurements on the dry matter. The authors

claim that during the maturity period, they monitored higher water needs, the time that the basil plant was developed, rising in that way dry matter and fresh values. During the harvest period (beginning of senescence) was observed around 3.16 mm/day consumption in water (63.2 mm for the whole period). In comparison to the previous phenological period, the authors claimed that they reached 48.1% decrease in water. The drawback is that the dry matter measurements approached 38.1% in comparison to the maturity period. As someone can understand the experiments shown in the current chapter are targeting on basil irrigation on specific hours with precise water.

The authors in [41] show the results of their experiments that took place on basil plants on how the temperature affects the plants. What they discovered is that by changing the temperature from 17°C up to 23°C, just 5 degrees more, there were more flowers in 'Sweet Dani', 'Lime' and holy basil. Researchers in [42] showed that by changing air temperature from 15°C to 25°C there are more flowers in Salvia splendens and Tagets patula (marigold). They also claim that when the temperature is more than a specific threshold, there are less flower, something better known as heat delay. The authors of another research [41] saw reduced reproductive values in 'Lime' lemon basil and 'Sweet Dani' lemon basil when the temperature was more than 35°C. As another research [43] states that there are 3 types of crops that have to do with the plant T_b:

1. Cold-tolerant crops, when the T_b is lower than 4°C.
2. Intermediate crops, when T_b is between 4°C and 7°C.
3. Cold-sensitive crops when T_b is more than 7°C.

The authors result in that a T_b should lie in the range 10.9°C to 12.1°C if it stands for a basil with fresh weight accumulations. For this reason, the basil is classified as cold-sensitive crop. In cases where the temperature is more than T_b, the implementation rate rises up to T_{opt}, and next it decreases down to T_{max} [44]. In general, it is a good tactic for plants to cultivate in temperatures ranging between T_b and T_{opt}. The latter has different values for different species.

In the following research [45] the authors made experiments on Sweet Basil specie, with small temperatures between houses that were fanned and houses that were not fanned. However, they found great differences in Relative Humidity (RH), about 95%. The RH, in the houses with fan, was nearly saturated in high frequency and maintained that for greater amount of time compared to houses with fan. This phenomenon has consequences to the sporulation of P. belbahri, as it makes it guide to more conductive environments. Only 4 hours are required for the basil to be infected in its leaves. When considering sporulation, the minimum time is 7.5 hours of intense relative humidity. The previously discussed conditions took place frequently in non-fanned houses, but they almost did not exist in houses with fan. The result of all these is the quick implementation of epidemics in houses without fan, in comparison to slow progression in houses with fan.

1.5 Experiments with the Arduino and its connected sensors

The experiments that took place in the basil pot, were consisted of the Arduino MEGA 2560 R3 and the following parts: 2 capacitive soil moisture sensors, air humidity sensor, air temperature sensor, UV light sensor, Xbee Zigbee Rx/Tx module. The experiments took place on a Ocimum Minimum variety basil.

An IoT ecosystem was built in the laboratory environment, with the aim to provide the necessary information to the user (farmer) to decide if the irrigation is needed or not, saving resources (water) and money. Aiming at the environmental conditions, such as: temperature, UV radiance, humidity, and soil moisture, the related sensing modules have been adjusted so that they send more measurements when the irrigation of the basil takes place. The latter helps to have a more precise view of the time of irrigation (before and after the event) and set the optimal thresholds for the soil moisture as it will be presented latter in the current chapter. All the data, as well as DSS (Decision Support System) and actuation took place remotely and wirelessly from the Raspberry Pi SBC, that was operated as a PC in reality.

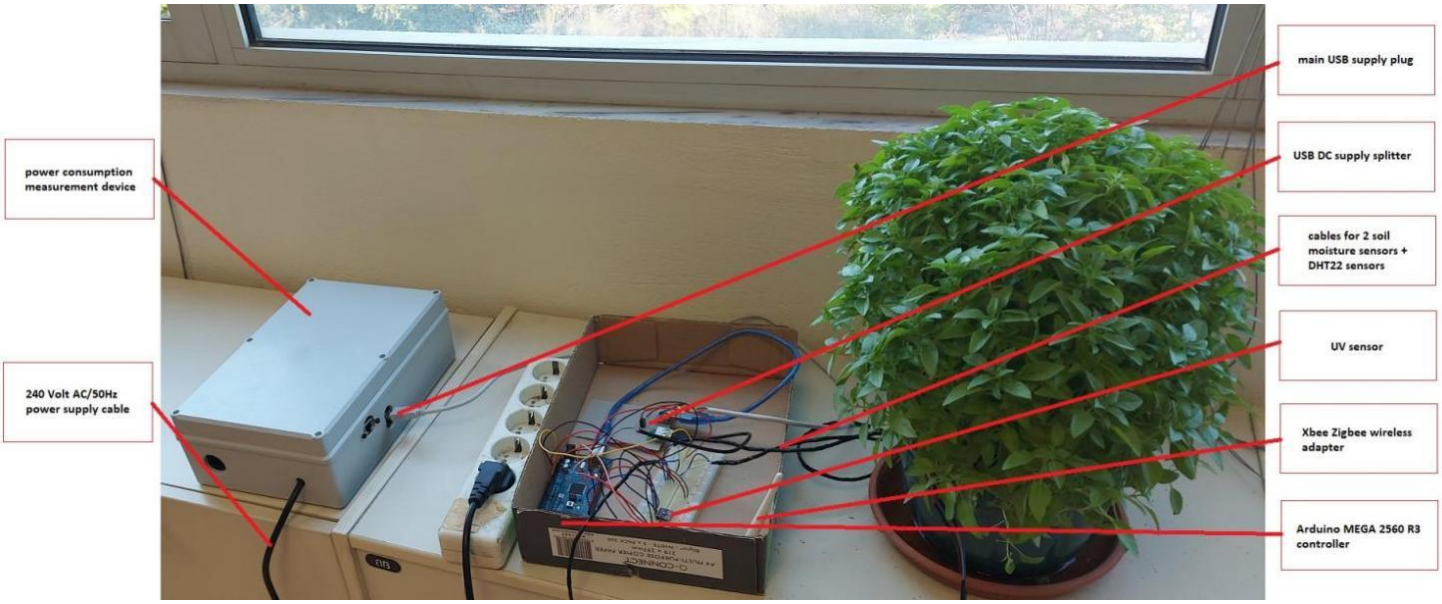


Figure 3 View of the infrastructure for sensing the parameters of the basil pot (temperature, humidity, UV radiance) and the measuring device, which measures and data-logs Voltage, Current and Power consumption.

The proposed constructure with the basil pot, the Arduino micro-controller, the various sensors and the measuring device are depicted in **Figure 3** View of the infrastructure for sensing the parameters of the basil pot (temperature, humidity, UV radiance) and the measuring device, which measures and data-logs Voltage, Current and Power consumption. The 2 soil moisture sensors, and the temperature/humidity sensor are placed inside the soil of the basil pot and on the basil's leaves respectively. However, the UV sensor is placed on the breadboard, which does not affect its measurements, since the light exists in all the space. The Xbee Zigbee is also placed in the breadboard and it is the main unit that is related to the data exchange with the other Xbee Zigbee connected to the Raspberry Pi. The main constructure is powered by the measuring device built by the author[8] in order to have datalogging of datetime, voltage, current and power consumption. All the data gathered by the Raspberry Pi, via the wireless connection between the 2 Xbee Zigbee, were displayed in a 17" TFT (Thin-Film-Transistor) monitor in real-time, and provided in order to support decisions on whether to irrigate or postpone the irrigation according to the values of the measurements. The data were stored in a .csv format file.

[8] The experiments have taken place in the (AIL) Ambient Intelligence Laboratory of the School of Electrical and Computer Engineering of the National Technical University of Athens in Greece.

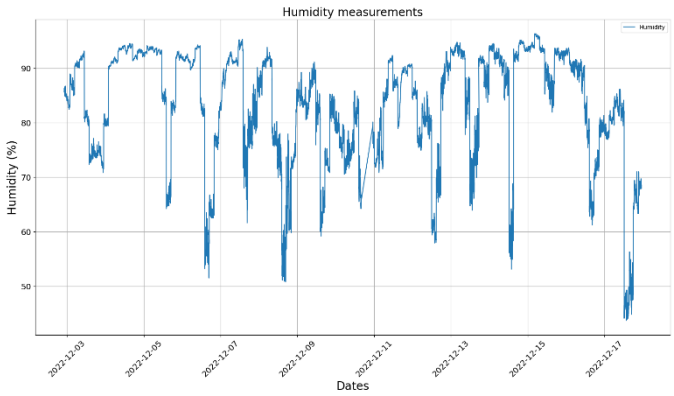


Figure 4 Relative Humidity (%) that was measured in the experiment. It is easy seen that humidity stays low in the day and rises in the night.

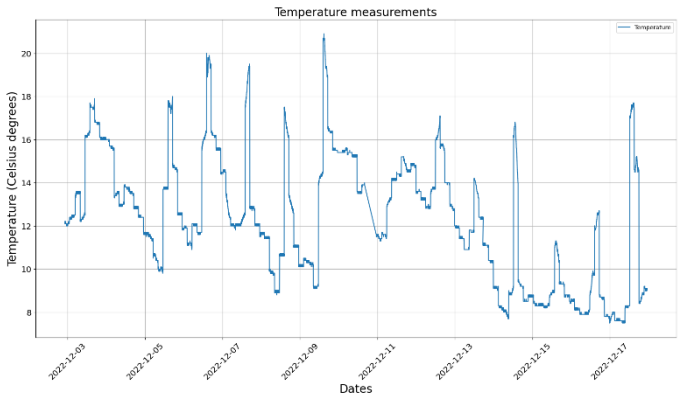


Figure 5 Temperature (ºC) that was measured in the experiment. It is risen in the day and decreases in the night.

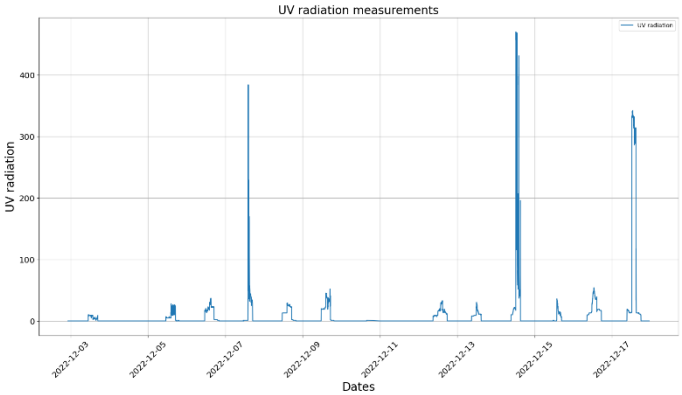


Figure 6 UV radiation that was measured in the experiment. It makes sense to have non-zero values during the day, where there is plenty of light and non-zero values at night.

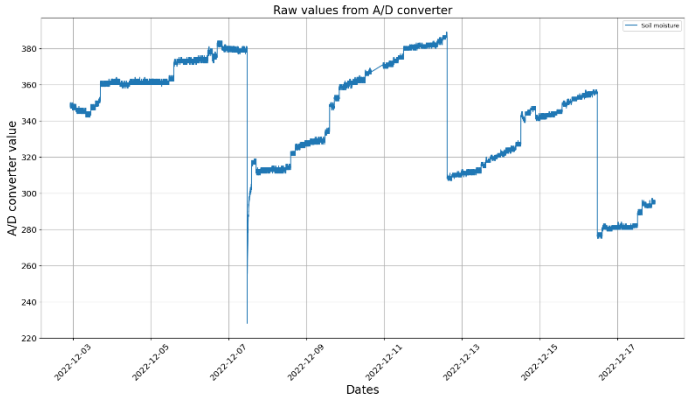


Figure 7 Soil moisture indirect calculation via the raw data coming from the A/D converter of the Arduino during the experiment. The higher the A/D converter's value displayed in this figure the lower the actual soil moisture. When irrigation takes place "negative spikes" are observed, whereas when dry soil exists, the values are increasing.

Many experiments took place in order to evaluate the best values and set them as thresholds to start or stop irrigation procedure. The experiments were realized between 03-12-2022 to 17-12-2022, two weeks in total. The frequency of the measurements was set to one minute. The measurements are depicted in **Figure 4**, **Figure 5**, **Figure 5**, **Figure 7**.

The way that capacitance soil moisture sensor operates, is very easy to understand. They translate their capacitance, as a result of the soil moisture in the ground, to voltage value which is then read by the Arduino MEGA 2560 via its A/D converter. The more increased the sensor's value, the drier the soil is. In **Figure 7** someone can view the indirect soil moisture values of the whole experiment via raw values read by the Arduino's A/D converter. The lower the raw data the higher the soil moisture and vice-versa. Before there is need for irrigation the values rise up to 360-390, next they fall very quickly when the basil pot is irrigated. After many experiments guide to the following numbers (thresholds) which indicate need for irrigation: 300-320. As it is displayed in **Figure 7**, three irrigations took place throughout the experiment. The DSS regarding irrigation is

not based only on that, but, also in humidity/temperature of the air and UV radiance. The measured current consumption was in the range 166 to 186 mA at 5.1 Volt. Using the equation P = V * I, someone can indicate power consumption in the range: 850 mWatts to 950 mWatts.

In **Figure 8** someone can view the UML (Unified Modelling Language) diagram of the proposed DSS idea. The Arduino microcontroller checks continually the sensors to identify if any of the set threshold (one threshold for each sensor). This operation is called polling. When the threshold is exceeded, the device irrigates the basil using 250 ml of water (the size of a tea cup in volume). There exists the Irrigation Delay Counter (IDC) which is by default set to the value "72". The latter means "72 hours". That number is decreased by 1 every hour, until it reaches 0. When the microcontroller meets the value 0, it starts polling round again. From the UML diagram it is more than clear that no sensor can trigger the system at the same time with any other(s) sensor(s).

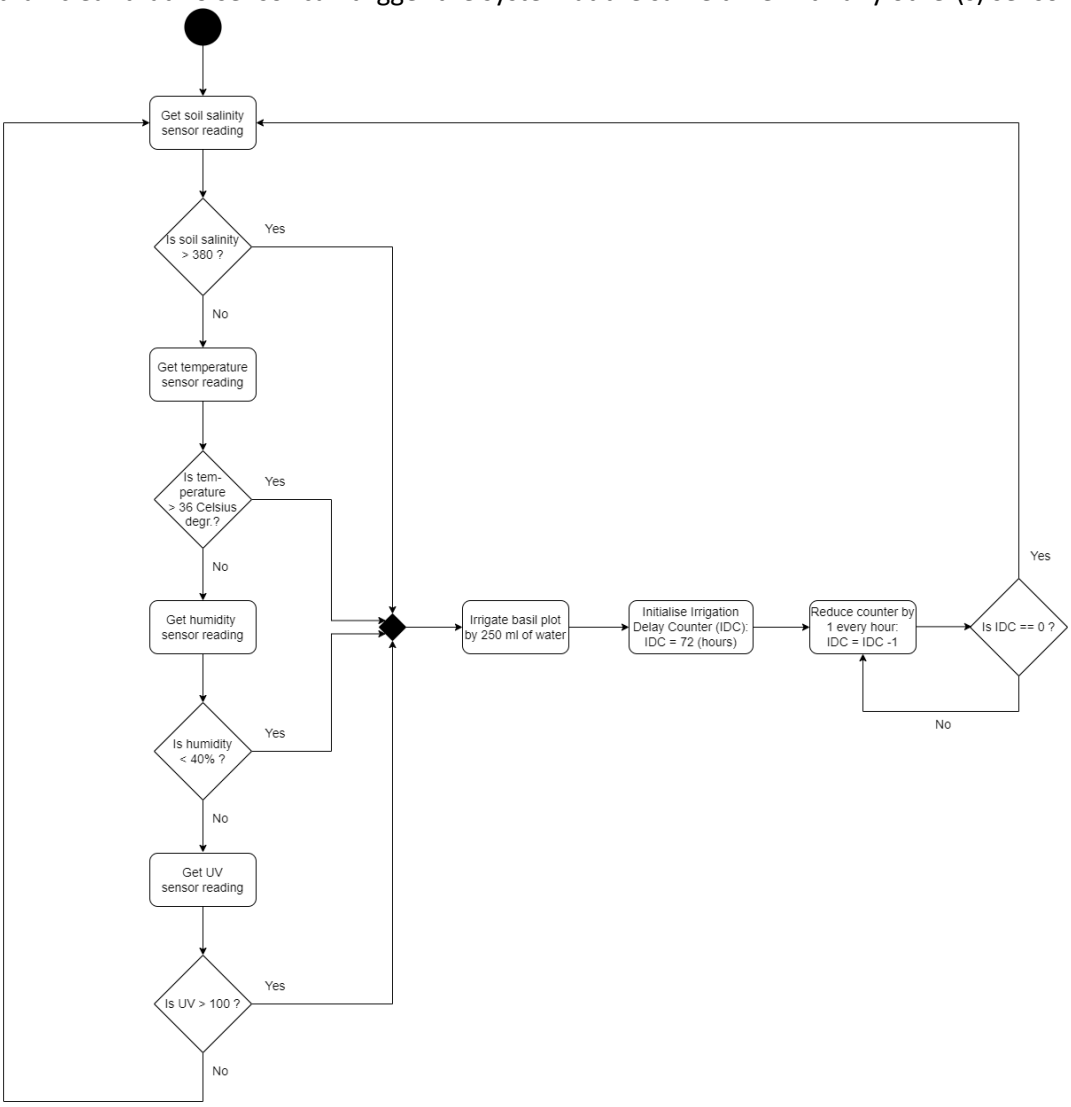


Figure 8 The UML diagram of the DSS (Decision Support System) proposed in the current chapter.

1.6 Savitzky-Golay Filtering

In order to somehow get signals concerning soil moisture values without “spikes”, so that be easy to identify the thresholds (upper and lower), Savitzky and Golay filtering [46] was used. There was use of *p*-degree polynomial for every continuous subset of *2m + 1* points, with $p \leq 2m$. Knowing that *0*-th differentiation stands for smoothing and *d*-th belongs into the range *0* to *p*. The latter exists in the mean point of the starting data and extracting the fitted polynomial that undergo differentiation. Then, polynomials regarding least-squares can be implemented by convoluting the data in the input. The last occurs by using a *2m + 1* digital filter. The coefficients of the convolution can be gained for any differentiation order, meaning all points in data and any degree of the related polynomial [46]. The negative issue, is that the latter does not exist for even amount of data. It exists for odd values only. In the related device that the current chapter is analyzing, Savitzky-Golay filtering has been applied in soil moisture measurements so that it could be easier to identify where to put the limits that irrigation is enabled or disabled. For instance, as the **Figure 9** depicts, when soil moisture value exceeds 280, the plant should be irrigated.

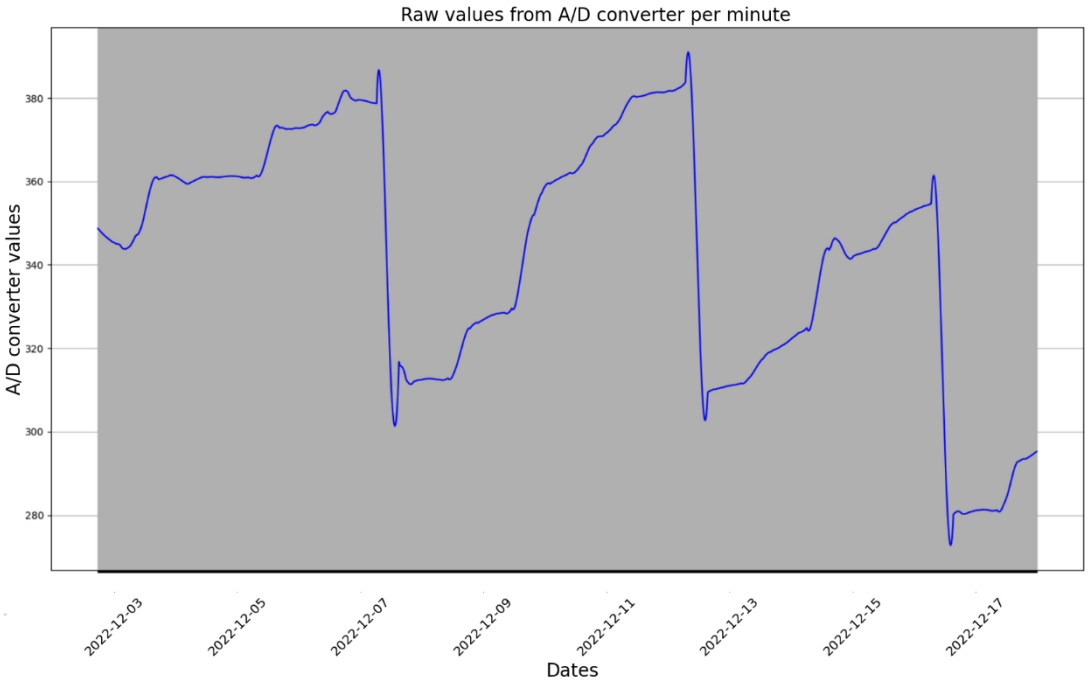


Figure 9 Indirect soil moisture from A/D converter raw values of the Arduino filtered with Savitzky-Golay filter in order to have an overall clearer picture of where the actual thresholds start and at which point, they end.

1.7 Introduction to RNN-LSTM neural networks

In [47] the researchers make an exhaustive analysis on the theory behind RNN-LSTM neural networks.

1.7.1 Recurrent Neural Network model

In the RNN neural networks, every neuron consists a processing unit which is attached to the output of its node at the input. Each neuron applies an activation function, before the output takes place. The neural networks because of the latter have the capability to construct nonlinear relationships. But the generalized neural model is not able to simulate the time parameter. All the

datapoints are constructed from fixed-length vectors. And when strong correlation takes place via the input phasor, the model can diminish the consequences of the processing. Recurrent Neural Networks (RNN) have the ability of modelling time. The definite time cannot take place into the output only, but can appear in the next time step layer, which is hidden, by putting time points coming from the hidden layer as well as the hidden feedback connection.

The mainstream neural network does not bring any middle layer of the process. The specialized input $x_0, x_1, x_2, \ldots, x_t$, after the process of neurons there will be a related output $h_0, h_1, h_2, \ldots, h_t$. In every training, there is no need from transfer of information between the neurons. The difference between RNNs and common (mainstream) neural networks, is the fact that in each training for the RNN, neurons need to bring some information.

The essential structure of the RNN is depicted in **Figure 10**. And in **Figure 11**, someone can see the deeper analysis. Someone can see that *A* stands for the hidden layer, x_i is the input vector and h_t stands for the output of the hidden layer.

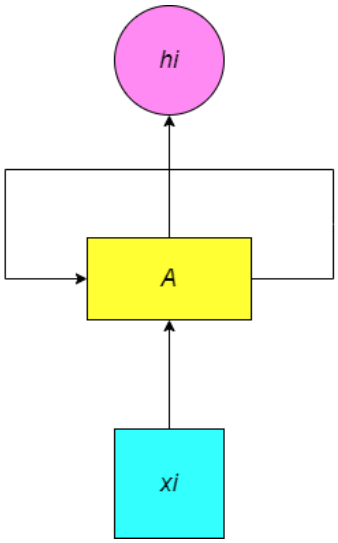


Figure 10 RNN basic model.

As someone can see in **Figure 11**, the output of every hidden stage is fed as input to the following layer.

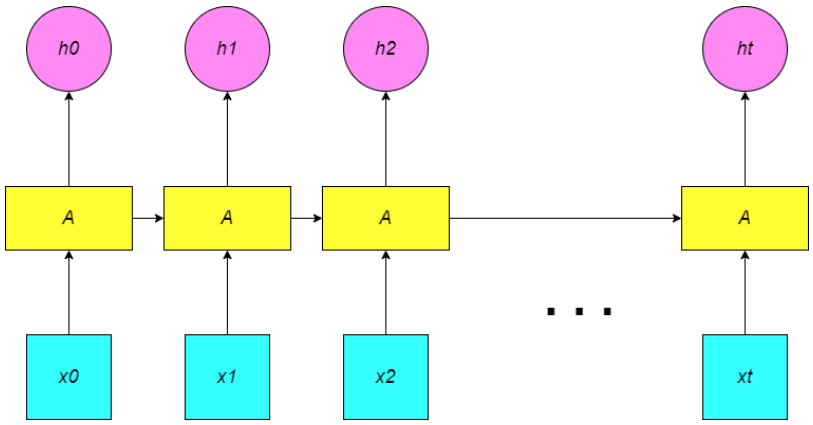


Figure 11 RNN expanded model.

1.7.2 Long Short-Term Memory Module

Although RNNs can handle non-linear time series, there are issues concerning gradient when training long time lags, which are basic in time-series forecasting. RNNs face also issues with predetermined time lags in order to identify temporal sequence computation and find the most capable time window size automatically. Thus, in order to excel such kind of issues that RNN face, LSTM-RNN were invented [48].

As it occurs with RNN, LSTM is also a memory module. It consists of 4 different essential units as it is depicted in **Figure 12**and **Figure 13**. These are the following:

i. an input gate
ii. a neuron with a feedback connection
iii. a forget gate
iv. an output gate

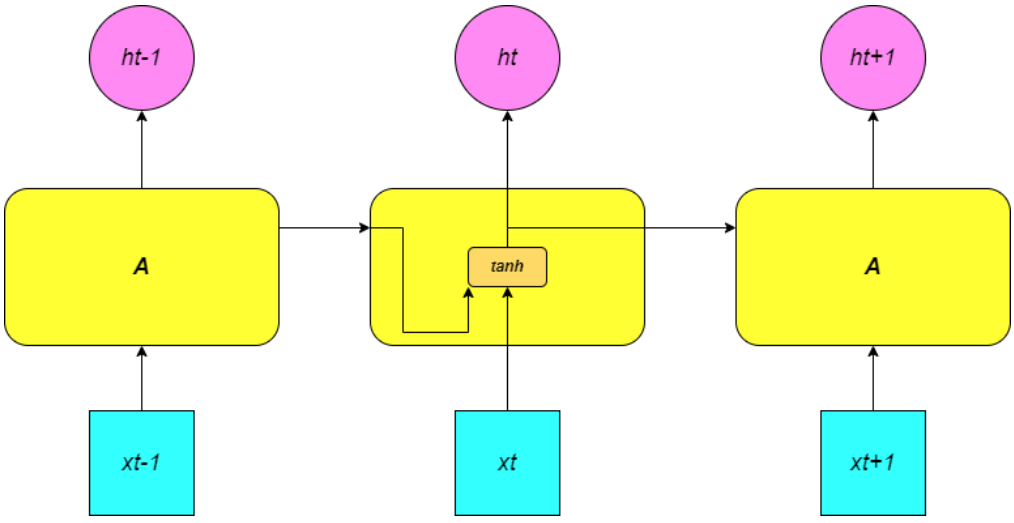


Figure 12 This image depicts the expanded single node of the RNN.

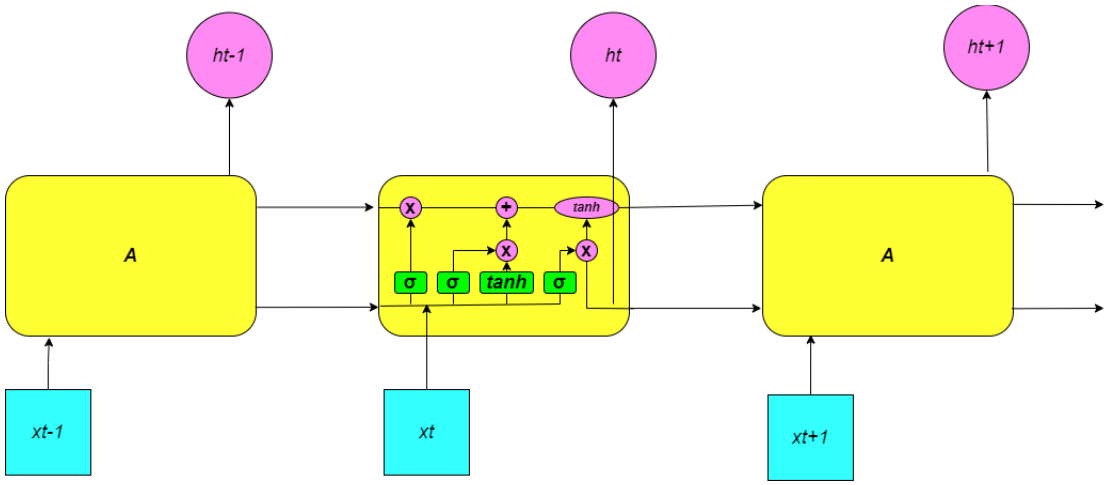


Figure 13 This image depicts the LSTMs cells. As it is obvious, each LSTM cell contains four layers that interact.

The three nonlinear gates depicted in **Figure 14** are the unit that realizes the summation. The latter is responsible for controlling the inside and outside transmission of the information either through activation modules via the operation of multiplication. The multiplication exists at every input and output cell by their related gates. The forget gate multiplies the memory cell's feedback connection and lets the cell either forget or remember its previous state. That occurs via the realization of the sigmoid activation function. The f_t activation function gate is assumed to be a logistic sigmoid, so that gate activation belongs to the range 0 (stands for gate close) to 1 (stands for gate open). The *tanh* or logistic sigmoid lets the output activation function, O_t, in order to excel the vanishing gradient issue. Its second derivative can be maintained for a long range before resulting to zero. Further growth is needed, which is based into a different problem statement. The weights connect the cell to the various gates, as is depicted in **Figure 14**.

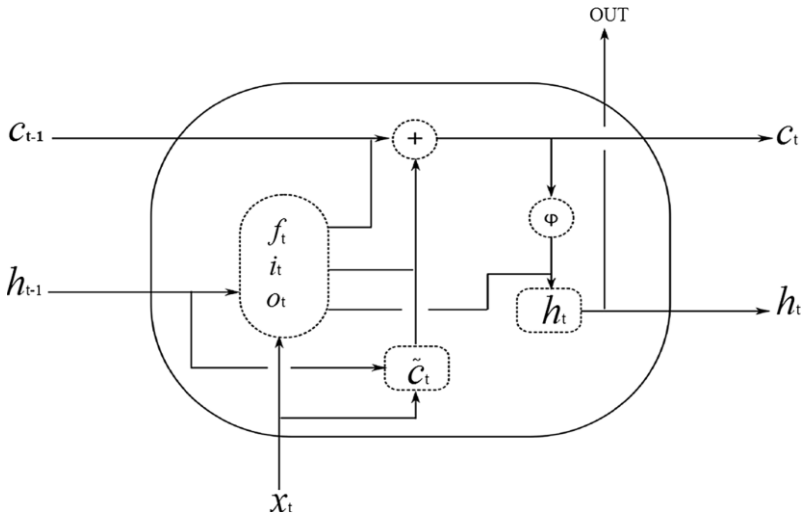


Figure 14 Representation of the LSTM memory block, which consists of one cell with 3 gated layers [48].

More increase is feasible that is based on the different problem statement. The various weights connect to the gates, as it is depicted in **Figure 14**. The rest connections are without weight. The memory module links the rest of the network via output gate multiplication.

The model input is given by the formula:

$$x = (x_1, \dots, x_j, \dots, x_t)$$

And the output sequence is given by the following formula:

$$y = (x_1, \dots, x_j, \dots, x_{t+t'})$$

where *t* stands for the prediction and *t'* stands for the next time step prediction. The *x* can be seen as a historical input data, and *y* as a single lag in period series. Both the latter in the case that low-flow takes place. The main target of LSTM-RNN is to forecast low-flow discharge in the following step based on the former data. All these are calculated from the following equations:

$$i_t = \sigma(W_{ix} \cdot x_t + W_{ih} \cdot h_{t-1} + W_{ic} c_{t-1} + b_i)$$

$$f_t = \sigma(W_{fx} \cdot x_t + W_{fh} \cdot h_{t-1} + W_{fc} \cdot c_{t-1} + b_f)$$

$$c_t = f_j \cdot c_{t-1} + i_t \cdot g(W_{cx} \cdot x_t + W_{ch} \cdot h_{t-1} + b_c)$$

$$o_t = \sigma(W_{ox} \cdot x_t + W_{oh} \cdot h_{t-1} + W_{oc} \cdot c_t + b_0)$$

$$h_t = o_t \cdot h(c_1)$$

$$y_t = W_{yh} \cdot h_t + b_y$$

Where σ stands for the sigmoid function. The memory is schemed in a box and contains an input gate, an output gate and the forget gate. They are given by the following: i_t, o_t, f_t. The symbols c_t and h_t stand for the cell and memory block. The symbols *W* and *b* represent the weight and bias vectors respectively, in order to generate a linkage between the output layer and memory block.

1.8 Forecasting basil pot conditions using RNN-LSTM

The proposed device of the current chapter elaborates on an RNN-LSTM neural network, where the code is written in python. It can output forecasts for air temperature, air humidity, UV radiance and soil moisture. As it is depicted in **Figure 15**, **Figure 16**, **Figure 17**, **Figure 18**, someone can observe the actual values, the prediction on trained data and the prediction on tested data. The different datasets are colored as following: red color indicates the real measurements, blue color indicates prediction on trained data, and finally green color indicates prediction on tested data. The dataset used consisted of 20.000 measurements and 400 epochs for training. This model can be used aiming at forecasting the soil moisture of basil pot and estimate if it needs irrigation. The rest of the parameters such as air temperature, relative humidity and UV radiance can be used secondarily in order for the user to decide when to irrigate in the future. The prediction on the unseen (tested) data is very accurate, thus the small error as it is presented in the following **Figure 15**, **Figure 16**, **Figure 17**, **Figure 18**. The ML model gets an input of time-series and it is able to decide, or better, to forecast when it is the appropriate time to irrigate. The ML model identify the pattern and then can easily forecast. What is new about the current infrastructure is that it

can successfully forecast the various conditions and it is able to decide when to irrigate saving water resources and not spending them when it is not needed.

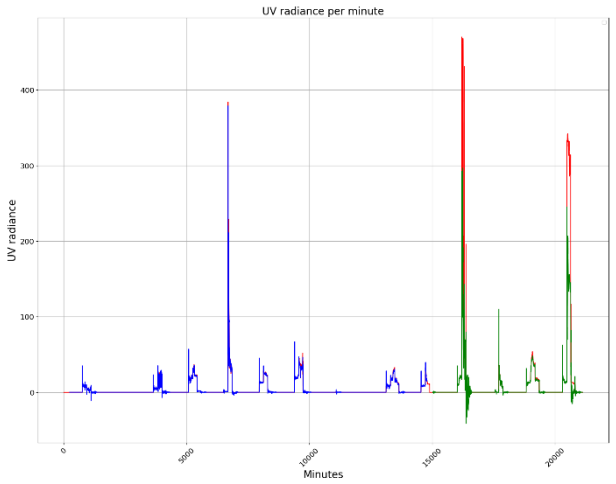


Figure 15 UV radiance data-logging with 1 minute frequency and 20.000 values in total. The actual (initial) dataset is colored with red, predictions on trained (known) data are colored with blue color and predictions on unseen (unknown) data are colored with green.

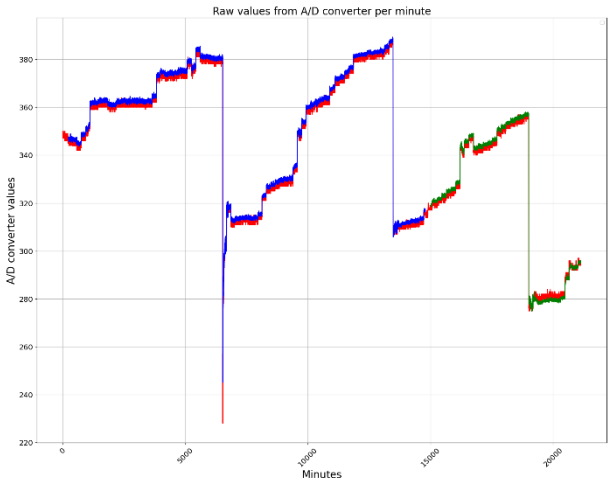


Figure 16 Indirect soil moisture data-logging with 1 minute frequency and 20.000 values in total. These are raw values coming from the A/D converter of the Arduino. The actual (initial) dataset is colored with red, predictions on trained (known) data are colored with blue color and predictions on unseen (unknown) data are colored with green.

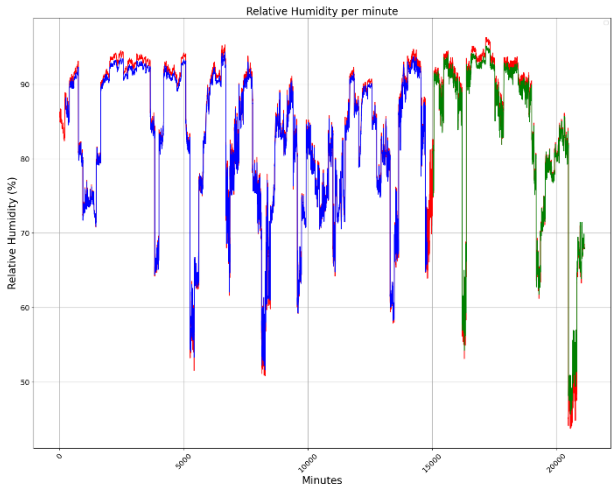


Figure 17 Relative Humidity data-logging with 1 minute frequency and 20.000 values in total.

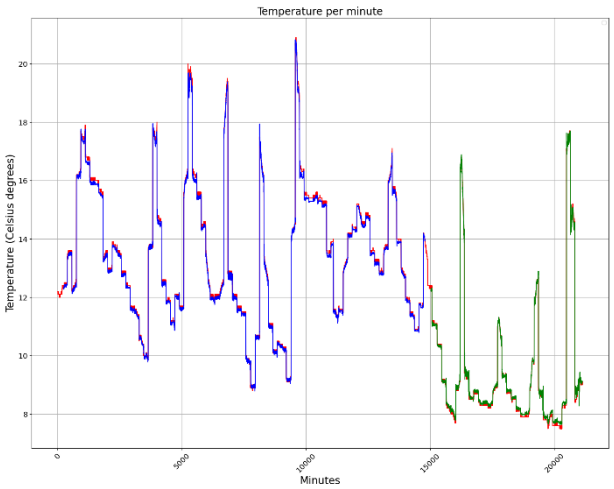


Figure 18 Temperature data-logging with 1 minute frequency and 20.000 values in total.

The building of Machine Learning models leads to identify if they work correctly and accurate. In order to understand the previous, specific metrics should be provided. For this reason, seven metrics were calculated:

- Root Mean Square Error (RMSE)
- Mean Square Error (MSE)
- Mean Absolute Error (MAE)
- Squared (R^2)

- Correlation Coefficient (CC)
- Relative Absolute Error (RAE)
- Root Relative Absolute Error (RRSE)

More analytically, RMSE is connected to the SD (standard deviation) of variations or divergences between the forecasted values and the values that were measured. MSE can be considered the same rationale as RMSE without the squaring. MAE is related to the absolute values, differing between forecasted and measured values when best try takes place, without considering the sign of the values, but estimating the prediction error series. R2 outputs (shows) the reliability of the regression algorithm, aiming at making easier the changed values. As far as CC is concerned, it evaluates how accurate is the ML model, by remaking the outputs used in experiment. RAE is about dividing the whole absolute error with the whole absolute error of the main indicator. RRSE squares the RSE, and more specifically it provides normalization of the whole squared error through the division with RSE total square error.

	RMSE	MSE	MAE	R^2	CC	RAE (%)	RRSE (%)
Humidity	0.82	0.68	0.45	0.99	1.00	0.06	0.08
Soil_moisture	1.75	3.07	0.79	1.00	1.00	0.04	0.07
Temperature	0.16	0.02	0.06	1.00	1.00	0.03	0.07
UV radiance	3.68	13.55	0.24	0.91	0.95	0.04	0.31

Table 3 Metrics related to performance for sensors reading evaluation targeting training part.

	RMSE	MSE	MAE	R^2	CC	RAE (%)	RRSE (%)
Humidity	1.43	2.04	0.69	0.99	0.99	0.07	0.11
Soil_moisture	3.09	9.57	1.98	0.99	1.00	0.08	0.11
Temperature	0.42	0.17	0.27	0.97	0.99	0.17	0.18
UV radiance	13.95	194.49	2.2	0.95	0.98	0.09	0.22

Table 4 Metrics related to performance for sensors reading evaluation targeting testing part.

During the testing and training periods, metrics calculation took place. The results are depicted in **Table 3** and **Table 4**. Both Tables show the 4 parameters sensed by the sensors: humidity, soil moisture, temperature and UV light. The values depicted are not high, so, for instance, MAE takes values ranging from 0.06, concerning temperature MAE, to 0.79, concerning soil moisture MAE, on the training part. As far as the testing part is concerned, the values start from 0.69 for temperature MAE to 2.2 for UV light MAE. As it is obvious, the smaller the value of MAE is, the more well the ML model operates. Analysing the CC metric, it takes a minimum value of 0.91 to a maximum value of 1.00 for any parameter, both for training and testing measurements, something that indicates a high correlation between real and predicted values. R^2 shows the same correlation as occurs with CC. The RRSE takes the value of 0.17 in average in the range of estimations. It is also obvious that RMSE and MSE are low for the following parameters: soil moisture, humidity and temperature, but not on UV light, which are high on both periods (testing and training). The reason for the latter is the "spikes" that exist in the measured values.

In order to identify how good or bad the RNN-LSTM used on forecasting time-series: it is very crucial to use loss functions. These functions are depicted in **Figure 19**, **Figure 20**, **Figure 21**. **Figure 19** demonstrates the loss on relative humidity dataset, **Figure 20** depicts the soil moisture loss function and **Figure 21** shows the temperature loss. Orange color is used to display validation loss and blue is used for train loss. The training procedure had the following configurations: ADAM (Adaptive Moment Estimation) optimizer, 400 epochs, with 0.1 Dropout. What someone can understand from the curve shape is that the used ML model works really well.

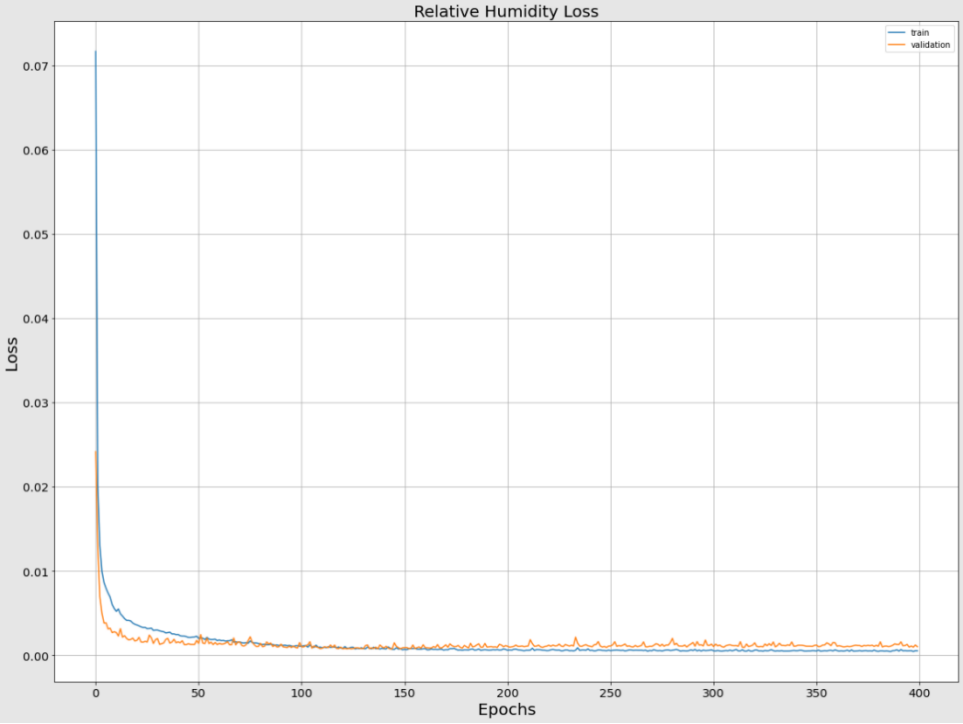


Figure 19 Relative Humidity Loss function operating for 400 epochs training, both for validation (with orange color) and test (with blue color) values.

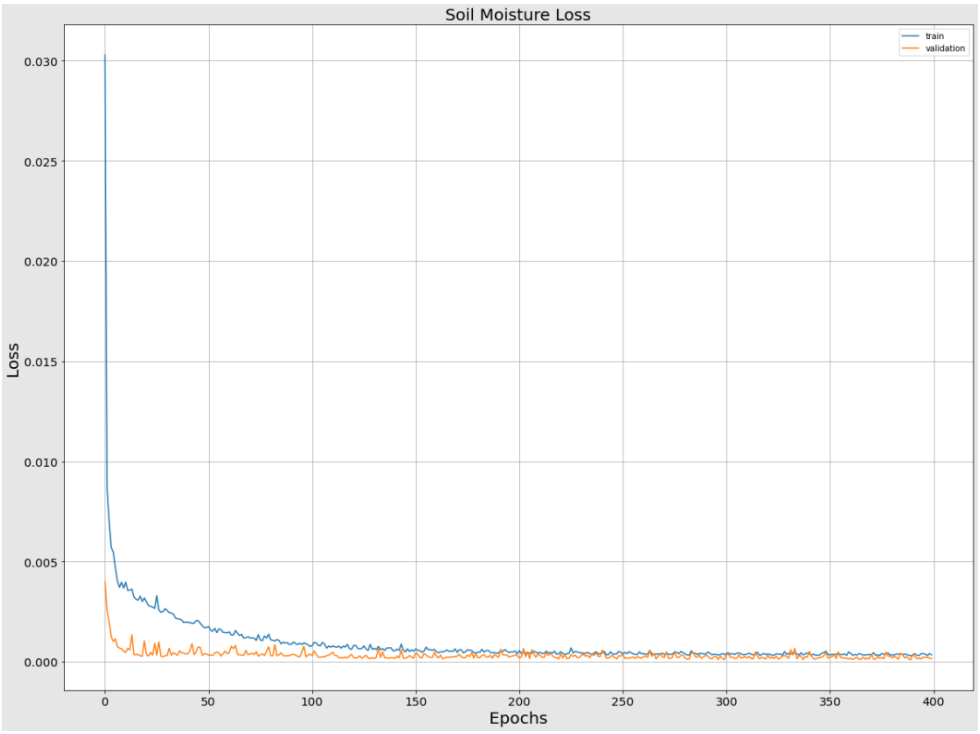


Figure 20 Soil Moisture Loss function operating for 400 epochs training, both for validation (with orange color) and test (with blue color) values.

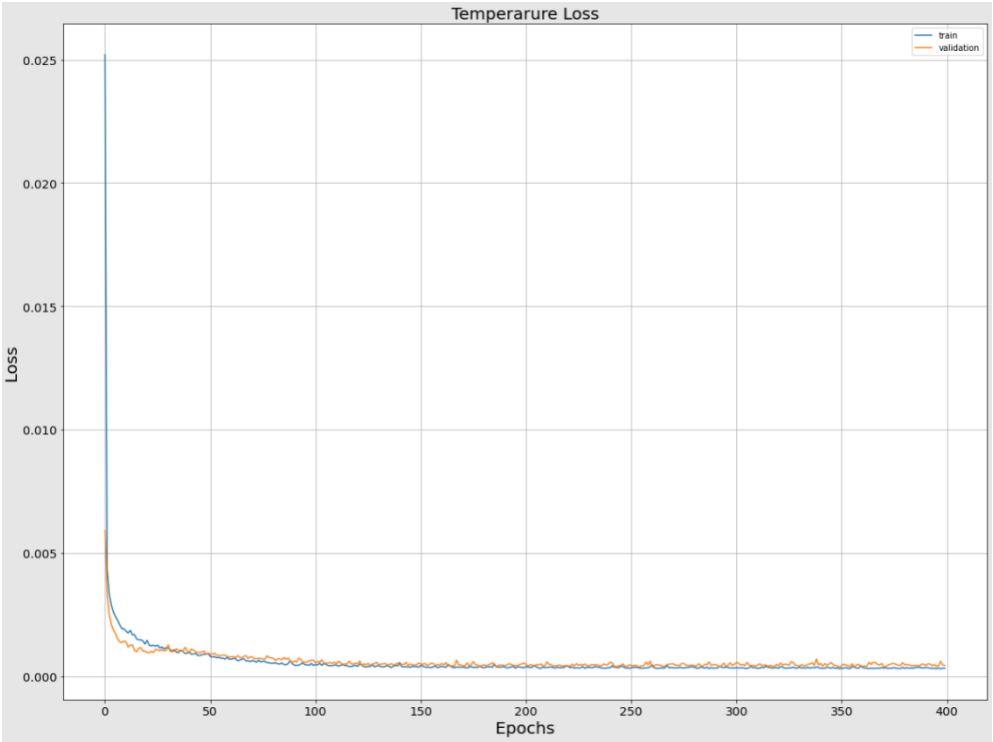


Figure 21 Temperature function operating for 400 epochs training, both for validation (with orange color) and test (with blue color) values.

1.9 Issues related to power consumption, control and monitoring

The main issue when using IoT devices is the energy consumption. The idea is that such type of devices or modules have to communicate with micro-controllers with reduced battery, CPU, bandwidth resources, so it is very critical to use the right modules to have long time of operation with long-lasting batteries. Below there is the analysis of the IoT device, concerning issues like energy consumption, control and monitoring.

The experiment with the basil pot took 2 weeks, continues operation 24/7. Below in **Figure 22**, **Figure 23**, someone can see how the power consumption was allocated throughout the experiment. In **Figure 22**, it is depicted the power consumption with 1 hour frequency. So, the data were gathered every 1 hour. In **Figure 23**, someone can see how the power consumption was allocated with frequency sampling equals to 1 minute. The power consumption ranges from 875 mWatts to 950 mWatts approximately. There are some intense minima ("negative spikes"), indicating that at these timestamps maybe there is less information sent from the Arduino on the basil pot to the Raspberry Pi aggregator. The result is smaller messages, so less power used by the Zigbee modules.

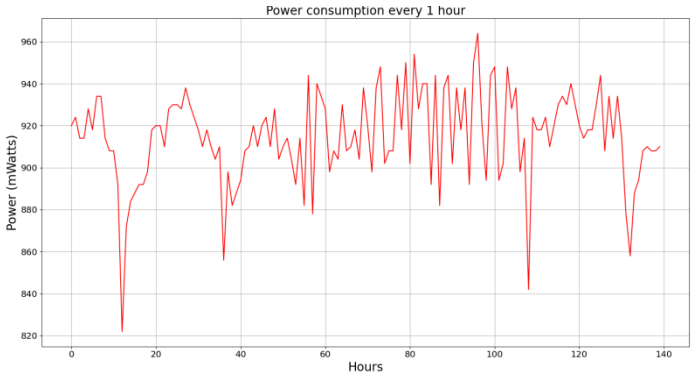


Figure 22 Power consumption of the completed circuit, measured in mWatts, with 1 hour sampling frequency.

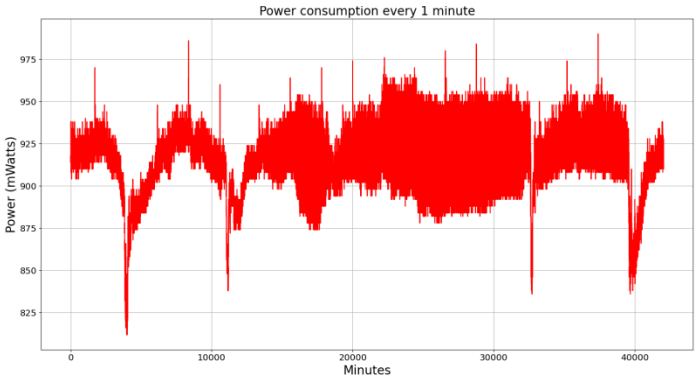


Figure 23 Power consumption of the completed circuit, measured in mWatts, with 1 minute sampling frequency.

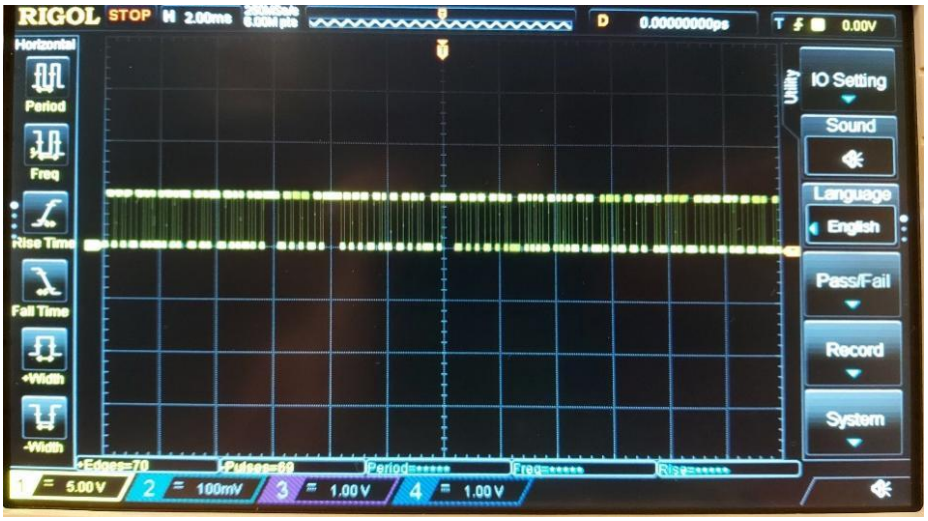


Figure 24 In the picture someone can see how is a signal (message) that is sent from the Arduino (existing in the basil pot) to the Raspberry Pi, via Xbee Zigbee is viewed under the oscilloscope. The current needed from the Xbee Zigbee module is around 40 mA, according the manufacturer[9]. The whole device operates at 5 Volts, which results in a power consumption of P = V * I = 5 * 40 mA = 200 mW on peak of the signal.

In order to estimate power consumption, an oscilloscope was used. One of the 2 Zigbees was connected to the oscilloscope, while messages were sent from the Arduino to the Raspberry. One such power imprint is depicted in **Figure 24**. The parameters seen are the following: Ts = 2 ms/DIV (Division) and V = 5 Volts/DIV. The peak consumption can be calculated from the well-known formula: P = V * I = 5 * 40 = 200 mWatt approximately. If there is need to calculate the energy consumption in Joules, there are more that should consider, such as the duty cycle, and using Ts and P someone can easily calculate the energy.

For the energy consumption research on the wireless modules used on the current area, an extensive analysis takes place. For that reason, 10 different modules were tested in the Lab[10]. The 10 different Schemes are depicted in **Table 5**. They use the same sensors as they are displayed in the table, but different wireless modules and 2 different micro-processing modules. Scheme 1, uses 2 soil moistures sensors, drawing 4.8 mA each one, a humidity/temperature sensor and a UV light sensor, one Arduino MEGA 2560 R3 for processing all the data from the sensors and one Xbee Zigbee S2 module for the Tx/Rx of the data. From measurements on the modules, the current draw was 164.3 mA. Scheme 2 contains the same modules as the Scheme 1 but as a radio module instead of Xbee Zigbee S2, it uses the famous SIM900 GPRS module, with the whole current draw of the Scheme to be 433.30 mA. Scheme 3 used the same modules as Scheme 1 with different radio module, so, it uses SIM7600E 4G modem, and the measured current draw was 747.3 mA. Comparing those 3 schemes it is obvious that GSM modem need more current. Continuing with Scheme 4, it uses the same elements as Scheme 1, but different wireless module: instead of the Xbee Zigbee, it uses LoRa radio module at +13 dBm, with needed current measured at 174.3 mA. The same scheme as the Scheme 4, with more an amplified LoRa module at +20 dBm is used form Scheme 5 and it needs 275.3 mA. Schemes 6, 7, 8, 9, 10 are identical to Scheme 1, 2, 3, 4, 5, but instead of Arduino as the main processing micro-controller they use the Raspberry

[9] https://www.adafruit.com/product/968

[10] The experiments have taken place in the (AIL) Ambient Intelligence Laboratory of the School of Electrical and Computer Engineering of the National Technical University of Athens in Greece.

Pi 4B without any fan on it. The results were measured as follows: Scheme 6 needs 345.3 mA current, Scheme 7 needs 614.3 mA current, Scheme 8 needs 928.3 mA current, Scheme 9 needs 355.3 mA current and Scheme 10 needs 456.3 mA current. As someone can understand, Scheme 1 is the least power-hungry circuit needing only 821.5 mWatts, whereas Scheme 8 is the most power-hungry circuit needing 4641.5 mWatts. All these are depicted in **Table 5**.

	Operating current (mA)									
Element	**Scheme 1**	**Scheme 2**	**Scheme 3**	**Scheme 4**	**Scheme 5**	**Scheme 6**	**Scheme 7**	**Scheme 8**	**Scheme 9**	**Scheme 10**
Soil moisture sensor	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8
DHT22 sensor	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
VEML6070 UV sensor	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
Arduino MEGA2560 R3 (measured without pins used)	109	109	109	109	109					
Xbee Zigbee	41					41				
SIM900 GPRS (EGSM 900) mean of (PCL=5)		310					310			
SIM7600E 4G (20Mbps)			624					624		
Adafruit RFM96W LoRa Radio (+13 dBm)				51					51	
Adafruit RFM96W LoRa Radio (+20dBm)					152					152
Raspberry Pi 4B						290	290	290	290	290
TOTAL current consumption (mA)	164.3	433.30	747.3	174.3	275.3	345.3	614.3	928.3	355.3	456.3
VCC (Volts)	5	5	5	5	5	5	5	5	5	5
Power consumption (in mWatts)	821.5	2166.5	3736.5	871.5	1376.5	1726.5	3071.5	4641.5	1776.5	2281.5

Table 5 View of the various Schemes, their connected sensors, their micro-controllers or microprocessors, and their different power consumption (in mWatts).

According to research papers that are referred below, the **Table 5** is verified. More specifically in a study by [11] they used a GPRS modem for data communication that needs 200 mA current and it is more than the Xbee Zigbee used in the current chapter, which needs 41 mA. In another study [12] the authors support that the Rx/Tx modules that consume the least power are BLE and Zigbee. Indeed, BLE is even more less power consuming than Zigbee, however, the distance a BLE covers is about 10 meters. Zigbees can reach 120 meters at LOS (Line-Of-Sight). Authors of another study [49] they use WiFi modules to transmit data from IoT agriculture-based devices to the (Rajkumar et. al., 2017) Cloud, consuming between 1150 mWatts to 1300 mWatts, far more than the device of the current chapter, which uses about 821.5 mWatts, as shown in **Table 5**. Lastly, in another research [13] they claim that Zigbee modules for transmitting/receiving wireless data need low current.

1.10 Conclusions

The current chapter presented a novel device related to smart irrigation mechanism that was based on IoT rationale, such as Arduino module, as the main processing unit, different sensors attached to it, Zigbee wireless module for the communication with the data logging device (Raspberry Pi 4B). The outcomes of the performance are quite satisfying due to the fact that the total mechanism construction was very straight forward and the real-time results of the devices, as well as their computation needed low resources in order to trigger end-users being aware of compact status/condition data and references. The irrigation process has been proposed in order to let the users change to their needs the wished operation frequency based on the crop needs for water. An RNN-LSTM Machine Learning scheme was proposed that can make forecasts related to UV radiance, temperature, relative humidity and moisture in soil, in order to help the DSS (Decision Support System). The new added value of the current chapter is the lightweightness of the current mechanism, which is easy to build and change its parameters, it shows low power consumption, and the significant high accuracy of the RNN-LSTM scheme.

We have started working on a mechanism for maize farms in Northern Greece, where an AI (Artificial Intelligence)-enabled DSS that takes as input not only data from the various sensors that they were described before, but also data from UAV/drone/satellite images. The target of that scheme is to mitigate the water usage, while at the same time preserving the health of the plants.

Moreover, there is an effort to combine UAV and satellite images as inputs, in order to process RNN-LSTM algorithms targeting on delivering better quality irrigation related forecasts, and be more accurate on how much water is needed in the plants. The whole mechanism is planned to be waterproof, thus being an advantage, since all the previous described devices are not. Finally, the device will be supplied by solar panels in addiction to batteries, so they will be working on any weather conditions.

This page was intentionally left blank.

Chapter 2: Plant diseases identification using Single Board Computers (CPU, GPU, TPU) and Machine Learning models

2.1 Introduction

The basic issue that the current chapter elaborates on is the image processing with the related classification of images with leaves, according to whether they are diseased or healthy. In order to extend a previous classification dataset consisted of 10- 15 classes, it was decided to realize the problem with 33 classes for the processed leaves, with an effort not to destroy the accuracy of the model. The used dataset was gathered from a publicly available repository[11]. The dataset used lacked balance considering the number of the images. So, pre-processing took place, in order to calculate the initial number of images. The number of images in the class was 152. In every class were kept 3 * min = 456 images, while some classes had more than 1000 images, that would have negative effects on the accuracy of the model, since the learning phase would be grounded on them. The pre-processing was realized in the Cloud via the auxiliary use of Google Drive. The training of the whole scheme was implemented on Google Colab, a tool that makes good use of GPU and TPU architecture, in order to accelerate Machine Learning code, written in python programming language. Then the trained model was uploaded to Single Board Computers in order to implement the prediction on unseen (new) images. Machine Learning algorithms were executed in Single Board Computers, such as: Raspberry Pi 3B+, Raspberry Pi 4B, NVIDIA Jetson Nano, Google Coral TPU Edge Dev Board, which all of them belong to the IoT technology. The aim was to produce a model using as less as possible resources, such as low RAM, and decreased CPU power. The pictures were loaded every time in ML model and a random filtering was implemented on them in order to change various parameters such as the range of colour: in a range 0-255 value, brightness, zoom, etc. The idea on these processes is to have as more realistic dataset as it can be, because the pictures that would be fed by the user will not be in perfect condition, and so the ML model should take those imperfections in consideration. So, the model should handle cases where the picture is rotated or not, do not have the suitable lighting, etc. In IoT devices it is very crucial to consume as little energy as possible, because there are power constraints, especially if the SBC operates with the support of a battery along with a solar panel, or a small wind generator. Apart from metrics concerning RAM, CPU, temperature, time, other measurements took place concerning energy consumption. The latter was implemented with a special measuring device that was constructed in the lab[12]. The device data-logged voltage (in Volts), current (in milli Amperes) and power (in milli Watts) of the load.

[11] https://github.com/spMohanty/PlantVillage-Dataset
[12] Ambient Intelligence Laboratory, National Technical University of Athens

2.2 Related state-of-the-art

2.2.1 Machine Learning models used in the agri field

The current sub-section demonstrates the various research papers on Machine Learning models in the agriculture field, and it is not limited to devices consuming low-power.

In [50], the authors present a device they have built consisting of the following components: DH sensor, in order to capture humidity and temperature, LDR (Light Dependent Resistors) for measuring light, soil moisture sensors and the famous NodeMCU for wireless communication. Concerning the software, it included Firebase, Jupyter Notebook, python 3.5+ programming, a text editor (atom sublime), Flutter framework and dart language, Ngrok localhost webhook tool for developing and the well-known Arduino IDE. The core of the mechanism is an Intel R CoreTM i5 processor 8300H clocked at 2.60 GHz/2.80 GHz (1 socket, 4 cores, 2 threads per core), which uses 8GB DDR4 RAM and 2 GPUs interchangeably: HD (High Definition) graphics 630 or NVIDIA GeForce. The rationale of their scheme is that the API URL (Uniform Resource Locator) is sent to the ngrok, and the output is sent to the app in a JSON (JavaScript Object Notation) structure. The API is linked with label files that contain the diseases, the image converter, the CNN algorithm, which the users use in order to communicate with the app. More specifically the model proposed is consisted of the following parts:

1. The first one is the REST API, which mainly includes: the dataset with the plant leaves, the CNN algorithm, and the Django REST tool in order to construct the API. The authors made use of more than 9.000 images/10 plants leaves/13 sections. They used 200 images per category for training their CNN model and the rest 700+ images for testing their CNN model.
2. The second part, stands for the flutter construction of the application and the connection with the customized REST API. It is consisted of two parts: *(1)* the design of the APP and *(2)* the linkage of API with the APP.
3. The third part is connected to the field monitoring.

As they describe in their research, their application brings 80% accuracy for 10 samples of "potato early bright" and 80% accuracy for 15 samples of the "Tomato yellow Leaf curl virus", for 10 samples of the "Apple black rot" disease, for 10 samples of the "Grape Black Measles". It outputs outstanding results for the 15 samples of the "Corn common rust" where it hits 93.33% accuracy.

In another case [51] the authors have constructed a device which contained the following parts: soil moisture sensors, module for sensing humidity, module for sensing temperature, a water sensor that can be placed either to water tank or into pesticide tank, a driver for DC motors, a DC motor and a robot mechanism. A relay driver with a relay communicates with a sprinkler device. All those parts, that described above, exchange data with a Raspberry Pi, that is linked to an Android application. So, their mechanism works as following: an image depicting a disease is selected, and it is shown in the APP. The APP is built using python programming language. If the farmers identify a disease, they give signal to the sprinkler device in order to spray pesticides or fertilizers and also water. Their device makes use of single pole relay for enabling/disabling the various devices. A module that senses water level is used by the farmers. The following sensors are used: LM35 (temperature), DHT-22(humidity), water sensor, moisture sensor. The sequence they use in order to identify a disease is like this:

a) Image obtainment
b) Pre-processing
c) Segmentation
d) Extraction of various features
e) Classification

The researchers used 900 images containing cotton leaves. They separated them into 629 for training period and 271 images for testing period. For the disease “Bacterial Blight” they hit accuracy of 85.89%, for “Alternaria” disease they hit 84.61% accuracy, for “Cerespora” they hit 82.97% accuracy, for “Grey Mildew” they reached 83.78% accuracy, for “Fusarium Wilt” they reached 82.35% accuracy and finally for “Healthy leaf” they achieved 80% accuracy.

In [52] the authors make use of the famous Resnet-50 neural network. The leaves’ disease position was extracted via the use of Convolution layers. So, the disease classification was indicated by iterative learning. In order not to face overfitting, they used random data increment. A Leaky-ReLU function with a 11 x 11 size kernel was used, aiming to the network change. That selection enhances the network ability to indicate features with details and enhance the receptive phase. The authors reached a rise of 2.3% in the testing phase during the performance of the network. They used 3000 images of the three most common leaf diseases in their experiments, these are: a) yellow leaf curl, b) Spot blight, c) Late blight. The ratio on training and testing period was 9:1, so in absolute numbers: 2700 images for training and 300 for testing. Images were stored and classified in folder, where each folder had a relation with the name of the disease, so there was a label for each category. Resnet-50 was compared with many activation functions and convolution kernel sizes. Adam optimizer was used with 0.00001 weight decay. Twenty iterations took place for training results the model was stored every 100 iterations. Ubuntu 16.04 OS was used with an NVIDIA RTX2060 GPU for training, and tensorflow package in python programming language. Below are represented the results of their experiments:

i. Training accuracy hit 97.7% and testing accuracy reached 95.7% on a 7 x 7 ReLU, in 51 minutes.
ii. Training accuracy hit 98.1% and testing accuracy reached 97.3% on a 7 x7 L-ReU, in 53 minutes.
iii. Training accuracy gave 98.3% and testing accuracy reached 98.3% on a L-ReLU, in 54 minutes.

In [53] the authors propose a scheme for identifying apple leaf disease. Images of apple leaf were put in sections, by firstly shading the apples’ green sections as well as the background section, and only capture the areas of the pictures that contain the apple leaf. The spots include certain colors and special textures characteristics according to the various diseases. The authors followed the below steps:

i. They bring threshold division, and they delete the background.
ii. They delete the green color mask in order to collect the diseased leaves.
iii. They compute both the color instantaneous feature of the grayscale instance matrix and the textures.
iv. They make use of SVM (Support Vector Machine) neural networks for training the model.
v. They continue the computation with the last image and assess it by using SVM models.

The authors used images of apple leaves coming from four categories, three of them were diseased leaves and one category included healthy leaves, so, a sum of 2700 images. More analytically, 380 images depicted variants of “black star disease”, 180 images carrying “cedar

rust", 427 images related to "grey spot" and 1185 images depicting healthy leaves. All the experiments were made on Intel i5-8265U CPU clocked at 3.00 GHz, incorporating 8GB RAM and Windows 10, 64-bit, through the use of Python programming language and more specifically the 3.6.8 edition. In order to accomplish image processing (recognition) the researchers used Tensorflow v.1.12.0 executing the code in an NVIDIA GeForce GTX 1050Ti 3GB GPU. They used the free open-source and well-known "Plant Village" dataset. They separated the dataset into the healthy ones (1185 pictures) and the diseased ones (987 pictures). They kept a 6/4 ratio in training/testing part something that is analysed in 1276 images (60%) for the training phase and 850 images (40%) for the testing phase. An accuracy of 90% was reached via the use of SVM model and image segmentation. They made use of ResNet-18 and ResNet-34 for both training and classification aiming at making the ML model sturdier. Thus, the model increases its accuracy to 99% in ResNet-18 and 97% in ResNet-34.

2.2.2 Single Board Computers executing CNN ML code

In the current sub-section, there are examples of CNN algorithms executed on SBCs, which are not bounded only in agriculture sector. The rationale is to present experiments from the literature that use CNNs executed on limited resources IoT devices, and as a result the reader gets informed of how the CNNs behave on various domains in devices such as Raspberry Pi, NVIDIA Jetson Nano, NVIDIA Jetson TX2 and other devices. As a result, the current section analyses the execution of CNNs on SBCs in general and gives a feeling to the reader of the accuracy and configurations made in various fields accomplished by the various researchers.

In [54] the authors show performances of SBCs in NVIDIA Jetson Nano, NVIDIA Jetson TX2 and Raspberry Pi 4, through the exploit of a CNN model which was made by the contrast of fashion product images. 2D CNN model was developed so that they could classify 13 different fashion objects in tests. Their dataset contained 45K images. Various parameters targeting performance analysis was gathered as consumption in GPU, CPU, RAM, power, accuracy and cost also. Dataset was organized to parts of 5K, 10K, 20K, 30K and 45K for training and testing periods in order to demonstrate the differences of each Single Board Computer. They resolved the performance of each embedded SBC scheme in low power with limited resources hardware devices. More precisely, as they state, they reached 97,8% with a 45K dataset on the Jetson TX2.

In [55] the authors present a low-power CNN model so that they can accelerate tasks of edge inference of RTC systems, where all the operations take place in a column-wise logic and realize an instant computation for inputting data in its input. They show that most calculations of CNN filters can be applied and terminated in multiple cycles and do not react on the total latency of the many calculations. They propose a multi-cycle scheme in order to implement the column-wise convolutional calculations in so that they can diminish the hardware resources and the energy harvesting of the devices. They enter a hardware architecture for multi-cycle algorithm such as domain-specific CNN architecture which is fulfilled in a 65nm transistor technology. They state that their scheme realizes up to 8,45%, 49,41% and 50,64% power reductions in algorithms such as LeNet, AlexNet and VGG16. Their experiments present that their approximation outputs bigger power mitigation for the CNN models made of greater depth, larger filters and more channels.

In [56] the authors propose a real time system aiming at the surveillance via the use of Rapsberry Pi and CNN for recognizing faces. They use as input a dataset consisted of labels. They start by training the system on labeled dataset so that they can export various characteristics of the face

and key points of face recognition. Then it compares faces and outcomes a result based on voting. The classification accuracy of their mechanism is guided by the CNN algorithm and it is compared with the widely known HOG (Histogram of Oriented Gradient), and also the state-of-the-art face detection and recognition methods. Moreover, the accuracy of their mechanism is stretched in faces with masks or sunglasses or live videos and they assess them. They reach the following accuracy: 98% for VMU, 98,24% for face recognition and 95,71% in 14 celebrity datasets. The outcomes of the experiments present their proposed model in accurate face recognition in contrast to the modern identification and recognition techniques.

In [57] the researchers have realized and evaluated efficiency and performance of an embedded scheme based on CNN algorithm on the Raspberry Pi 3. Their CNN models are in charge of classification of dissimilarities between many frames that include healthy and failure conditions in the structure. They transacted experiments and evaluated the CNN model via the use of piezoelectric patches linked to an aluminum plate. They managed a hit rate of around 100%. The latter accuracy has a great influence in the concept of CNN-centered SHM (Structural Health Monitoring) systems where implemented applications are wanted in order to recognize many and different damages in the structure, with application fields varying from aerospace structures, rotating mechanisms and wind generators.

In [58] the researchers propose a lightweight CNN model, the WearNet, so that they can realize automatic scratch recognition for materials existing in metal forming. A dataset consisted of surfaces gathered from a cylinder-on-flat sliding tests was used so that they train the WearNet with appropriate configurations in learning rate, gradient algorithm and mini-batch size. An in-depth analysis on the network results and decision offer was also recognized to show the proficiency of the developed WearNet. The outcome was that in contrast to existing networks, WearNet realized an excellent classification accuracy of 94,16% containing smaller model size and less time duration recognition. WearNet excelled compared to other modern networks when public repositories were selected for network evaluation. There were positive outputs identified when detecting surface scratches in the process of sheet metal forming.

2.3 The description of the problem

If the plant diseases are not detected in an early stage, there is the danger of rising in the production cost in agriculture [59]. This shows that a monitoring system should exist with high frequency in order to detect early disease signs, before the disease covers all the farm plant. It is obvious that monitoring the whole farm plant is quite difficult. However, with today's technology and via the use of remote monitoring and Machine Learning models it is something that can be realized. The current chapter examines the execution of Machine Learning algorithms for plant disease identification running on Single Board Computers (SBCs).

Over the last few years, Artificial Intelligence has shown tremendous success with applications in many fields, increasing the need for data and more intelligent and complex processing algorithms. This phenomenon underlines the need as much efficient resources as possible, such as RAM, CPU, energy. As a result, Machine Learning is the capability that machines gain to process various tasks by deciding the output without a human interaction, based on the natural knowledge that humans provide them at the early stages. Artificial Intelligence provides support to many fields in order to

solve problems, for instance: Machine Learning, NLP (Natural Language Processing), image processing and many others. Machine Learning is a sub-category of Artificial Intelligence. It is consisted of Algorithms that can be enhanced without human intervention (automatically) based on experience. The current chapter delegates the case of supervised learning, where the user uses labels in the data that are fed into the ML model (as it will be demonstrated in a latter paragraph). With classification the ML model can categorize data inputs and outputs. The proposed ML model checks a number of instances which are part of specific categories and makes use of known labels to identify in which category a recent input does belong. So, the mechanism is trained in a way to be able to separate features based on the training dataset, which comprises of the input data. Then, a validation dataset is fed to the ML model. The relation between the input data and the output labels is known, so the ML model is able to evaluate the learning operation.

The validation part works as follows: the validation data are input to the ML model and are compared with the real values of the output. Next, is the learning phase where the user feeds the ML model with test dataset in order to have an assess of how accurate has the mechanism learned. In the last phase, the user covers/hides the labels from the ML model, however, the model classifies the input data with what it has learnt so far. At the end of the operation, it can calculate the number of instances that were correctly classified and thus, the model can be assessed for its reliability. Throughout the various experiments, Convolution Neural Networks (CNNs) where used, a solution for working with images and more specifically for classification problems. Classification is the operation of feeding the ML model with images and having it calculate a probability/percentage for each image to belong to each category.

2.4 Proposed Solution

Four different Single Board Computers were used so that they can process Machine Learning algorithms. These models are:

i. Raspberry Pi 3B+ 1GB
ii. Raspberry Pi 4B 4GB
iii. NVIDIA Jetson Nano
iv. Google Coral TPU Dev Board

Raspberry Pi 3B+[13] is a Single Board Computer which 64-bit CPU with four-cores clocked at 1.4GHz. It supports dual-band WIFI (2.4GHz and 5GHz), Bluetooth version 4.2 and BLE, faster Ethernet. It also supports PoE with separate PoE HAT (Hardware Attached on Top). It includes full-size HDMI, four USB version 2.0 ports and a 40-pin General Purpose Input Output header. The Raspbian OS that uses, is flashed in a 128 GB microSD card. The device is supplied by a 5Volts/2.4 Amperes power supply. Its CPU is being cooled via the use of a connected fan. Due to the many cores of its CPU, the Raspberry Pi 3B+ can process jobs in parallel logic, minimizing the time of the output in comparison to a single-core CPU.

An even more advanced version of Raspberry Pi was used, the Raspberry Pi 4B with 4G RAM[14]. It consists of a four-core processor, the Cortex-A72, an ARMv8 64-bit architecture clocked at 1.5 GHz, with 4 GB LPDDR4 (Low-Power Double Data Rate - 4), 3200 MHz SDRAM (Synchronous

[13] https://static.raspberrypi.org/files/product-briefs/200206+Raspberry+Pi+3+Model+B+plus+Product+Brief+PRINT&DIGITAL.pdf
[14] https://www.raspberrypi.org/products/

Dynamic Random-Access Memory). It uses the newest Raspbian OS in a flashed 128 GB microSD. It is powered by a 5 Volt/3 Amperes DC power supply via a USB-C cable. It supports both 2.4 GHz and 5 GHz IEEE 802.11ac WIFI, Bluetooth version 5.0, BLE protocol, a Gigabit Ethernet protocol. It can provide up to 4K60 output via its duo micro-HDMI. It also incorporates 2 USB version 3.0 and 2 USB version 2.0. It encompasses the well-known 40 GPIO header and PoE.

NVIDIA Jetson Nano[15] has small dimensions, but a very powerful GPU, that supports parallel processing of multiple thread neural networks, and can be implemented in areas like object detection, classification of images, segmentation and sound processing (speech processing). It consumes 5 Watts when operating, a very suitable device for IoT experiments. The OS it uses is a modified version of Ubuntu 18.04 Linux, for operating specially in the NVIDIA hardware. The GPU it incorporates, makes it special compared to Raspberry Pi, since it is ideal for parallel execution of code of applications related Neural Networks. Its dimensions are bigger than the Raspberry Pi, more specifically: 69 mm x 45 mm, and has a heatsink and a fan in order to cool the system. It has a 260-pin edge connector, its CPU is clocked at 1.43 GHz, with four cores in the CPU (ARM A57), its GPU makes use of 128-cores Maxwell, and also the RAM is a 4 GB 64-bit LPDDR4 with 25.6 GB/s. In order to connect with other devices, it includes many protocols and ports, such as Gigabit Ethernet, M.2 key E, HDMI port, 4 USB version 3.0, GPIO, I2C, I2S (Inter-IC Sound), UART, SPI (Serial Peripheral Interface). The whole device is powered by a 5 Volts/3 Amperes power supply.

Google Edge TPU Coral Dev board [16] is a circuit dedicated to specific application, better known as ASIC, that stands for Application Specific Integration Circuit. It was made by Google and operated in cases where there is need to execute Machine Learning algorithms which are executed very fast, using the interface of Tensorflow lite with very low power consumption. Inference is characterized as the period needed for the completion of a process for provision by making use a trained Machine Learning model. Google Coral is a device for general-purpose processes related to Machine Learning code. It uses the famous Linux Mendel OS, a Debian-based Linux edition. It incorporates NXP I.MX 8M SoC (four-core Cortex-A53, Cortex-M4F) CPU and a GC700 Lite Graphics GPU, and the key processor for ML models is the Google Edge TPU coprocessor, able to provide 4 TOPS (4 Trillion Operations Per Second), with a very low power consumption, equals to 0.5 Watt/TOPS equivalent to 2 TOPS per Watt. A well-known example is the process of the MobileNet v2 at about 400 FPS (Frames Per Second)[17]. The device includes 1 GB LPDDR4 RAM, 8 GB eMMC, a microSD placket, a MIMO (Multiple-Input and Multiple-Output) 2x2 version WIFI (802.11b/g/n/ac) which operates on both 2.4GHz and 5 GHz bands, Bluetooth version 4.2. It includes a type-C OTG slot, a type-A 3.0, type-C power, micro-B serial console and port supporting Gigabit Ethernet. The Linux Mendel OS is flashed to a 128 GB microSD card. The whole device is supplied by a 5 Volts/3 Amperes power supply.

As it is clear, the proposed approach does not use one SBC, but four with different main processing units. To be more precise, the Raspberry Pis use their CPU, the NVIDIA Jetson Nano its 128-core GPU and the Google Coral its TPU. Before there is the description of the experiments, it is crucial to analyze how the different main processing units operate.

[15] https://developer.nvidia.com/embedded/jetson-nano-developer-kit
[16] https://coral.ai/products/dev-board/
[17] https://cloud.google.com/tpu/docs/beginners-guide

CPU is used for general purpose works, and it follows the von Neumann architecture, something that means operating with memory and software[18]. CPUs consist very pliant units and this is their huge benefit. The user can load and execute any script he wants, including various applications. To give an example, a CPU can process simple spreadsheets, execute code related to robots, purchase online goods, control engines in a rocket, classify images using an ML model and many more. The negative issue with every CPU is the fact the hardware on which is implemented does not know a priori the calculation it will come next, but as soon as it read it. The CPUs are using registers, also known as L1 cache, so that they can store somewhere the results related to each calculation. The most well-known drawback of a CPU architecture is the von Neumann bottleneck. A CPU makes use of its ALU (Arithmetic Logical Unit), which is part of the processor, dedicated to arithmetic and logic operations executed on specific words understandable by the machine standing off as operands[19]. Moreover, it includes every type of functional sections such as: operational logic, register for storing data and sequential logic. The ALU of a CPU includes parts that can maintain and manage adders and multipliers, moreover only one calculation can be processed every time. As a result, there is the need from the CPU to access the memory, something that puts bounds on the throughput and consumes considerable amount of energy.

GPU is the essential mechanism in order to manage big amount of data in a computer capable of handling general purpose operations or scientific operations. This is a reality because of the high speed and performance that GPUs are encompassing in relation to huge quantity of data [60]. GPUs include the capability of utilizing high performance and speed in comparison to the mainstream CPUs, both in memory and computational perspective. The cost of GPUs has been decreased in a level that can be purchased by the average home user. Moreover, tools like CUDA have switched their goal so that they can be used in demanding computing tasks, and of course in applications related to general-purpose area. When the comparison between GPUs CUDA cores and CPUs comes in the foreground, it is well-known that GPUs can realize floating point operations in relationship to CPUs. As a result, GPUs can utilize very demanding parallel computations and succeed in achieving far better speed for certain applications, compared to CPUs. Desktops' CPUs basic rationale is centered around MIMD logic, something that is analyzed to Multiple Instruction Multiple Data. This means that every core operates independently to the rest cores, moreover for different operations, different instructions are realized. A good approach to exploit GPU's capabilities is to compile code using a programming language close to the processor, or very low-level, as they are more widely known, such as C/C++. The famous CUDA (Compute Unified Device Architecture) realizes and processes, most of the times, scientific operations on the GPU. GPUs excel CPUs when talking about performance, since they use SIMD instead of MIMD. SIMD is the known Single Instruction Multiple Data. What happens in practice, is that tasks are handled by more cores so that they can execute floating point operations that basically guide to a rising performance.

[18] https://cloud.google.com/tpu/docs/beginners-guide
[19] https://dl.acm.org/doi/pdf/10.5555/1074100.1074135

	Raspberry Pi 3B+	Raspberyy Pi 4B	NVIDIA Jetson nano	Google Coral Dev TPU Board
Performance	5.3 GFLOPs	9.69 GFLOPs	472 GFLOPs	4 TOPs
CPU	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz	ARM® Cortex® -A57 MPCore (Quad-Core) Processor with NEON Technology | Maximum Operating Frequency: 1.43GHz	NXP i.MX 8M SoC (Quad-core Arm Cortex-A53, plus Cortex-M4F)
GPU	Broadcom Videocore-IV	Broadcom VideoCore VI	NVIDIA Maxwell architecture with 128 NVIDIA CUDA® cores	Integrated GC7000 Lite Graphics
accelerator	/	/	GPU	Google Edge TPU ML accelerator coprocessor
Memory	1GB LPDDR2 SDRAM	4GB LPDDR4	Dual Channel | System MMU | Memory Type: 4ch x 16-bit LPDDR4 | Maximum Memory Bus Frequency: 1600MHz | Peak Bandwidth: 25.6 GB/s | Memory Capacity: 4GB	1 GB LPDDR4
Networking	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN| Bluetooth 4.2 | BLE Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN | Bluetooth 5.0| BLE	10/100/1000 BASE-T Ethernet | Media Access Controller (MAC)	Wi-Fi 2x2 MIMO (802.11b/g/n/ac 2.4/5 GHz)| Bluetooth 4.2 10/100/1000 Mbps Ethernet/IEEE 802.3 networks
Display	1 × full size HDMI MIPI DSI display port	2 × micro HDMI ports (up to 4Kp60 supported)	HDMI 2.0a/b (up to 6Gbps) | DP 1.2a (HBR2 5.4 Gbps) | eDP 1.4 (HBR2 5.4Gbps) | Maximum Resolution (DP/eDP/HDMI): 3840 x 2160 at 60Hz (up to 24bpp)	HDMI 2.0a (full size)
USB	4 × USB 2.0 ports	2 × USB 3.0 ports 2 × USB 2.0 ports.	4x USB 3.0| USB 2.0 Micro-B	USB Type-C power port (5 V DC) USB 3.0 Type-C OTG port USB 3.0 Type-A host port USB 2.0 Micro-B serial console port
Other	Extended 40-pin GPIO header	Extended 40-pin GPIO header	GPIO| I2C| I2S| SPI| UART	40-pin GPIO expansion header
Audio/Display	4 pole stereo output and composite video port	2-lane MIPI DSI display port 4-pole stereo audio and composite video port	Two independent display controllers support DSI| HDMI| DP| eDP: MIPI-DSI (1.5Gbps/lane): Single x2 lane | Maximum Resolution: 1920x960 at 60Hz (up to 24bpp)	Audio connections: 3.5 mm audio jack (CTIA compliant) Digital PDM microphone (x2) 2.54 mm 4-pin terminal for stereo speakers Video connections: 39-pin FFC connector for MIPI DSI display (4-lane) 2
Camera	MIPI CSI camera port	2-lane MIPI CSI camera port	12 lanes (3x4 or 4x2) MIPI CSI-2 D-PHY 1.1 (1.5 Gb/s per pair)	24-pin FFC connector for MIPI CSI-2 camera (4-lane)
Storage	micro-SD	micro-SD	eMMC 5.1 Flash Storage | Bus Width: 8-bit | Maximum Bus Frequency: 200MHz (HS400) | Storage Capacity: 16GB	Micro-SD| 8 GB eMMC
Power under load	5V/2.5A DC via micro USB connector 5V DC via GPIO header Power over Ethernet (PoE)–enabled (requires separate PoE HAT)	5V DC via USB-C connector (minimum 3A1) 5V DC via GPIO header (minimum 3A1) Power over Ethernet (PoE)–enabled	| Module Power: 5 – 10W | Power Input: 5.0V	powered by 2-3 A at 5 V DC using the USB Type-C power port
Multimedia	H.264| MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1| 2.0 graphics	H.265 (4Kp60 decode); H.264 (1080p60 decode| 1080p30 encode); OpenGL ES| 3.0 graphics	Video Decode H.265 (Main| Main 10): 2160p 60fps | 1080p 240fps H.264 (BP/MP/HP/Stereo SEI half-res): 2160p 60fps | 1080p 240fps H.264 (MVC Stereo per view): 2160p 30fps | 1080p 120fps VP9 (Profile 0| 8-bit): 2160p 60fps | 1080p 240fps VP8: 2160p 60fps | 1080p 240fps	4Kp60 HEVC/H.265 main| and main 10 decoder 4Kp60 VP9 and 4Kp30 AVC/H.264 decoder (requires full system resources) 1080p60 MPEG-2| MPEG-4p2| VC-1| VP8| RV9| AVS| MJPEG| H.263 decoder
Price	35 $	35 $	99 $	130 $

Table 6 Comparison of the different SBCs used in the experiments[20 21 22 23 24 25 26 27 28 29 30 31].

GPUs can maintain reliable data-parallel computation when there is low latency in communication information but increased compute/communication ratio. In general GPU RAM is considered as fast element, it copies data from HDD (Hard Disk Drive) (in the current chapter the proposed storing device is the SD card) very quickly because their capacity is not so large. NVIDIA Jetson Nano makes use of 4 GB RAM, something that covers the needs of the experiments that the current chapter delegates. General Purpose Graphics Processing Units, better known with the acronym GPGPUs, can process data using parallelism, as an alternative cost-effective solution. NVIDIA Jetson Nano [61] can be programmed via JetPack SDK (Software Development Kit) and optimized libraries targeting Deep Learning, Internet of Things, computer vision and embedded mechanisms. Via the use of CUDA cores, the programmer can realize a very capable development infrastructure for applications. Jetson Nano also includes a mix of GPU/CPU hardware aiming to push the system the code on the CPU part and speed up the complex part of the code to the GPU and the CUDA.

TPU is a kind of an accelerator that can be programmed. It is based on the linear algebra rationale that can support optimization in Machine Learning code[32]. This type of SBCs is not used, for instance, in on-line purchases, to control the engine of a rocket, or to control a robotic device, but is great for Machine Learning classification problems at an intense speed comparing to CPU, moreover, consuming less power and using constraint physical footprint. The crucial benefit of TPU in comparison to GPUs and CPUs, is the fact that it decreases the von Neumann bottleneck. The main job of a TPU module is the computation of matrices, also the rationale behind its design is the knowledge of each step of calculation is it can accomplish that computation. The engineers that designed the TPU, have input thousands of adders and multipliers and interconnect them in order to build a huge physical matrix via the use of the operators. This kind of architecture is named systolic array, and realizes the executions of the neural networks' computations. It operates as follows: Initially the TPU loads the parameters from its memory to the matrix of multipliers it includes, then TPU forwards the output to the following multiplier while it gathers the summation. The result is the aggregation of each multiplication output between parameters and data without any memory access taking part. So, a TPU can introduce intense calculation throughput linked to neural network operations through the use of low power consumption and small footprint [33].

2.5 The basics of Machine Learning

[20] http://web.eece.maine.edu/~vweaver/group/green_machines.html
[21] https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/
[22] https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/
[23] https://www.elektor.com/raspberry-pi-3-b-plus
[24] https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf
[25] https://magpi.raspberrypi.com/articles/raspberry-pi-4-specs-benchmarks
[26] https://developer.nvidia.com/embedded/jetson-nano
[27] https://www.waveshare.com/jetson-nano-developer-kit.htm
[28] https://coral.ai/docs/dev-board/datasheet/#features
[29] https://coral.ai/products/
[30] https://www.amazon.com/ELEMENT-Element14-Raspberry-Pi-Motherboard/dp/B07P4LSDYV
[31] https://www.raspberrypi.com/products/raspberry-pi-4-model-b/
[32] https://www.amazon.com/NVIDIA-Jetson-Nano-Developer-945-13450-0000-100/dp/B084DSDDLT
[33] https://cloud.google.com/tpu/docs/beginners-guide

With the rise of current state-of-the-art frameworks, Machine Learning and Deep Learning through the exploit of Convolution Neural Networks, great success takes place in work projects in the area of image processing and more specifically in image recognition [62] [63]. CNNs make up a successful area in the field of image classification, combined with deep learning techniques, and use of ReLU enabling functions, dropout levels and data augmentation. It is obvious that the more analysis occurs in such networks, the more processing power is needed and more daring the learning procedure becomes. But, with a few suitable configurations on the realization part the exploited resources can be optimized due to the necessity of being connected to IoT infrastructure. In the current chapter, the proposed approach does not use any high-power computers, however, low-cost SBCs are used, that include flexibility, consume trivial power compared to high tech computers or Clouds, and can execute many threads concurrently.

2.5.1 Convolution

Convolution is a calculation where 2 functions take part and the value indicates how similar are those 2 functions. For example, for *f* and *g* the convolution is given by the following formulas (for the discrete version):

$$f[n](0 \leq n \ \leq N-1) \text{ and } g[m](0 \ \leq m \ \leq M-1)$$

$$(f * g)[n] = \sum_{m=0}^{M-1} f[\mathrm{n}+\mathrm{m}]\, g[\mathrm{m}]$$

The procedure of convolution can be defined in 2 dimensions or even more, however, the current chapter delegates the 2 dimensions version, and this is applied in image processing. Below is shown a 2D-convolution between a filter and a data function, aiming at validating the likeness between them:

$$\text{Filter function: } F[r][s](0 \leq r \ \leq R, 0 \leq s \ \leq S-1)$$

$$\text{Data function: } D[h][w](0 \ \leq h \ \leq H, 0 \ \leq w \ \leq W-1)$$

$$\text{2D - convolution: } (D * F)[h][w] = \sum_{r=0}^{R-1} \sum_{s=0}^{S-1} D[h+r][w+s]F[r][s]$$

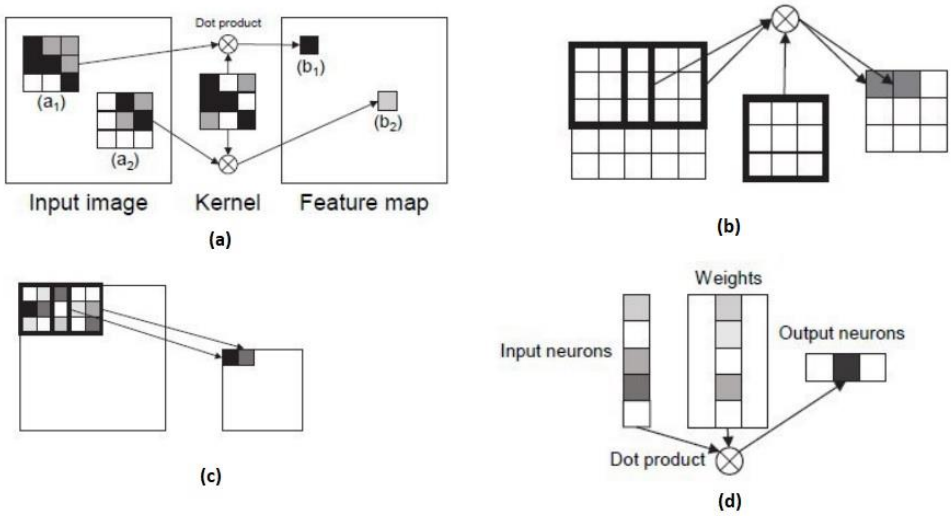


Figure 25 a) How the 2D-convolution looks like [62]. b) How Convolution with stride s = 2 seems [62]. c) A 3x3 max pooling with stride (step) s = 2 [62]. d) How a Fully Connected level looks like [62].

The 2-dimension convolution finds use in areas where there is need for image computation and is also called as image convolution. An image includes the data function $D[h][w]$ and accordingly the filter or kernel, as it is better known, includes the function $F[r][s]$. The output of the 2D convolution builds a feature map (matrix of the characteristics), whom role is to teach the program step-by-step, through the training, the essential characteristics of each image. The rationale is to learn, as good as it can, a set of images and then the Machine Learning model outputs which image set fits optimal (from what it collected in the input). Filters' size ($R \cdot S$) is smaller in comparison to the size of the image ($H \cdot W$). As it is depicted in **Figure 25**(a) for known parts of the starting image that is fed by the neural networks (a_1) and (a_2) that which have similar size to the filter function, they are then multiplied with the kernel filter. The outcome is a characteristics' map that includes the results (b_1 and b_2) of every multiplication. When the convolution gets high value, that means that the related chosen area of the image shows high degree of similarity with the filter [62].

The computation of convolution is very demanding in resources and it is more intense as the size of the image increases, because there is rise in the number of operations. There is possible sacrifice of accuracy, related to how the model learns the characteristics of the image, but there is gain in the time needed for the processes. The approached techniques are very mainstream while there is decrease in the operations without significant loses in the end result. The first solution is the implementation of 2D-convolution in the initial image with a stride (**Figure 25**(b)). This method results in down-sampling, meaning a decrease of the sampled image, because it collects strides every s pixels in every direction (horizontally and vertically). The s gives the value of the stride. Via down-sampling there is decrease in the image's parameters without significant loss of information. So, the output is compressed data in the feature map (output). The result is an image with smaller dimensions than the input.

The initial stage of the CNN is the part where the convolution computation is implemented in order to give the characteristics of the image related to the filter. Thus, the ML model is being taught during the process of training. The following stage implements the process of pooling that includes a down-sampling aiming to mitigate the time needed for computation and return useful

data that Neural Network will learn from them. In this stage a pixel representing a small area including pixels is chosen in order to decrease the input size. In **Figure 25**(c), someone can see the process of max pooling with step *s = 2*, which stands for an image divided in a 3x3 area using step 2, in every direction (horizontally and vertically). For every 3x3 area, only one pixel is selected, the one that has the highest value. The latter procedure is called max pooling.

CNNs make use of convolution filters, with step *s*, so that they can return the fed image's characteristics. The layers are more widely known as convolution layers, and the special thing about them is that between them pooling layers exist. As a result, down-sampling is realized and reduces the calculation size, returning significant information of the image. Convolution layer includes *N* filters and returns many characteristics of the given image, resulting in *N* feature matching. The part of the CNN that learns, it does so by one filter for every characteristic, targeting on recognizing it in other images.

2.5.2 Fully connected Layer

After collecting the outputs of the convolution layers and pooling layers, the next layer in a CNN model is the fully connected layers, that blends their results. As it is depicted in **Figure 25**(d) it includes the input neurons, the output neurons and the various weights. Its aim is to present the bonds between the input and the output. Moreover, it implements the multiplication between the matrix (storing the weights) and the input vector so producing the output. The logic of fully connected layers is to put the original image based on a label or class. In order to compute the weights, the fully connected layer of the CNN uses the idea of back-propagation in order to define the most precise weights. Each neuron aggregates weights and places in queue the most appropriate class or label where the given image is classified.

Summing the above, as it is clear the convolution layers are extractors of features. Convolution layers return maps of features, which characterize some area of the image input [63]. Each of the layers builds a tensor (according to the step) and gives input to the tensor of the next layer. Moreover, when all these layers end, the fully connected layers read the matrices (containing the last characteristics) and then flatten them. So, there is a transformation into one-dimension vectors and they make an output vector with *L* values, with *L* being the number of labels of the classes. At the final stage, a normalization method is implemented with the softmax layer, for *L > 2.* So, in every dimension the vector, which appears normalized, stands for the probability that the fed image matches/belongs to the adjective class of images.

All the steps, presented below:

- An image is fed to the Machine Learning model.
- There is realization of many filters in order to create the feature maps.
- Realization of the ReLU so that there is activation of the convolution layers.
- Flattening in a vector (one-dimension array), of the images after the last pooling layer.
- Feed the output of the flattening layer into the fully connected layer input of the CNN.
- Computation of defining the characteristics along the network culminates in the final fully connected level. The latter outputs a probabilistic distribution for the classes via the use of normalization.

- Training using forward and back propagation and many epochs, up to the point where there is well-defined neural network with enough trained weights and feature detectors.

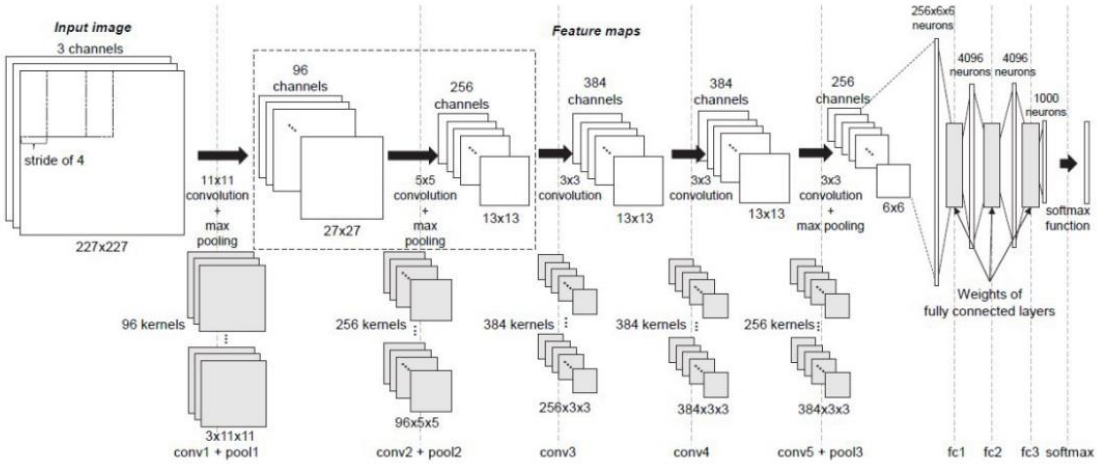


Figure 26 Scheme of the AlexNet neural network [62].

2.5.3 Softmax functions and ReLU

ReLU constitutes the activation function of every convolution layer, given by the formula:

$$y = max(0, x)$$

There is need of an activation function, which will trigger to learn many connections between the data, in order to make use of SGD (stochastic gradient descent) via the back propagation and also for training the neural networks [64]. The softmax function is fed with a vector of *K* numbers and indulges the latter using a probability distribution including *K* probabilities in relation to the exponents of the initial numbers. So, after the implementation of the softmax function, the entire number of values reside in the range *(0, 1)* and the sum of all values is 1 so they can be expounded as probabilities. Consequently, grander inputs return grander probabilities. The crucial softmax function which is used is given by the following formula:

$$\sigma: \mathbb{R}^K \to \mathbb{R}^K$$

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ for } i = 1, \ldots, K \wedge (z_1, \ldots, z_k) \in \mathbb{R}^K$$

The procedure of learning of a CNN follows the supervised process of learning via the use of images for training and the related correct labels that the latter images reside to. The inputs are supplied to the CNN per batch: per group of images. But there are signs that urge someone to optimize this part without deplete the memory, because many images are given in order to be processed. The ML model used in the current chapter realizes one batch each time because the SGD [65] mechanism, which is the mainstream approach in such kind of applications, has the capability to be easily work in parallel in images of the same batch.

2.5.4. Forward and Backward computations

Extending the analysis of the previously presented scheme related to the training, it is crystal clear that it includes a route that requests the most of the processing time, so it becomes very crucial for the CNN model to optimize time duration the best it can be. The idea of the current chapter proposed solution is a CNN model built from zero and after many tries it has been optimized, so as to solve the amount of data fed in its input.

CNN learning period is separated in 2 parts:

i. Forward computation, in which for an input image, the forward process proceeds throughout many levels of the CNN from the first to the last and gives back the result of the least layer.
ii. Backward computation, that gets the needed values that must be summed in the parameters of the network, for instance weights, of the ending fully connected level via the computation of the gradients.

When the update of parameters in the last layer finish, these values are brought to the previous layers and this is the backpropagation. As a result, the parameters can be configured or change in the backward processing level. The various gradients are computed with the logic of bringing the smaller difference between the output of last layer, here is the fully connected layer, and the actual value.

2.5.5 Connection between Machine Learning and agriculture

Machine Learning and especially Deep Learning is quite a new technique for image processing and contains many capabilities. It has been realized with significant success to many areas, and one such case is the agriculture. CNNs maintain a very serious role in applying many challenges corresponding to production in agriculture area. Something very interesting mention in the literature is the fact that success of a CNN model depends on a high degree on the quality of the data chosen. This is very crucial in the pre-processing stage, and will be analysed in a later paragraph of the current chapter. There are metrics used that are presented below:

- *Validation accuracy:* It shows the percentage of correct forecasting in the data (validation/test).
- *RMSE (Root Mean Square Error):* Usual deviation between predicted and actual values.
- *Precision, Recall, F1 Score:* Further in the current chapter precision and recall are analyzed. The F1 score stands for the harmonic average of the precision recall values. For multi-classification issues the F1 is computed in all classes.
- *Quality Measure:* It is computed by multiplying the sensitivity with the specificity. Sensitivity is calculated as the percentage of the correctly identified pixels and specificity meaning the percentage of identified pixels that are actually correct [66].
- *RFC (Ratio of the counted fruits):* It represents the ratio of the assessed number of fruits of a class as it is computed by the CNN model to the actual calculation that has foregone from the writers of by specialists [67] [68].
- *LC (LifeCLEF metric):* This has to do with the place of the correct elements in the list in the list that contains the recovered elements in the LifeCLEf 2015 Challenge [69].

2.5.6 Classification of images

The initial part for classifying images is that of pre-processing of data and dataset, a computation which is known as data pre-processing. The initial number of classes were 38 depicting leaves diseases. But there was a limited number of classes that included very few images or there was only a single depicting a leaf, so they were deleted due to the fact that they did not adduce something precious to the whole experiment. So, only 33 classes with diseases for leaves were maintained. It was observed that the class which contained the less images included only 152 images and other classes included more than 2000 images. This heterogeneity could output significant wrong results, if there was a try towards this training. Because of the fact that there would be total unbalance to the dataset and it could not learn to make a decision on related classes that include few images, so the files were uploaded on Google Colab and through the use of python scripts at maximum three times the number of images were maintained for every class. The latter was based on the number of images that were found in a class with the least images, that in this approach were 152 images in the class. With this rationale, the classes included at maximum 3 * 152 = 456 images per class. That happened because there was need to have a balance to all the classes and not affect the model with imbalanced classes. Then, the training part took place, along with the help of google Colab. In order to achieve this, there was used the google drive auxiliary to the Colab. Starting with, the Machine Learning model was built, in which there were configurations in issues such as the parameters of the layers. Concerning the accuracy part, it computes the number of rights estimations of the algorithm, in respect to the whole number of estimations. Two metrics, in addition to all the previous described metrics, were used. These were the Precision and Recall, and the target is to approach the unit (1) as much as possible. So, the extra metrics are the following:

$$precision = \frac{tp}{tp + fp}$$

where:

- *tp* represents the *true positive* and equals hit,
- *fp* represents *false positive* and equals false alarm

The other extra metric is the following:

$$Recall = \frac{tp}{tp + fn}$$

where:

- *fn* stands for *false negative* and is equal to miss

The loss function represents the categorical cross entropy, and it is used along with the softmax function. The latter is triggered in the last Dense layer. Via the use of this loss function the ML model outputs a probability for every image for belonging to each of the labels. When there are many classes to be classified, the labels are one-hot, thus each image has 1 in the label that suits, and 0 to all others. Only this value can be used for the loss computation.

Because embedded IoT modules were used, there should be a solution in order to decrease the RAM usage. If all the images were used at once at the beginning of the processing, there would be a rapid increase of the RAM needed. Thus, the *ImageData Generator* was used. The solution of data augmentation was followed by increasing the available data and not their number. In all

the training epochs, after the initial epoch, all images were transformed randomly, as happens to a common image that can be captured by an ordinary camera and could have horizontal rotations, vertical rotations, could have a slope, differentiation in brightness, channels could have been shifted, maybe have zoom etc. So, those differentiations in an image approach better a common captured image and the ML model with various imperfections can include all these and not learn just from perfect supplied images.

Due to the generator's capability the images were loaded into the ML model in batches, which resulted in conserving memory by computing smaller groups of images every time. For example: 8, 16, 32, 64 per time. When the processing was finished the images were removed from the memory.

Next was the period of fitting the model by following the validation loss. The latter loss is the loss for the validation set, that was described in a previous paragraph. It estimates the train of the ML model. The epochs were configured to 27 epochs with the following results:

i. validation loss: 0.3256
ii. validation accuracy: 0.9001
iii. validation precision: 0.9235
iv. validation recall: 0.8596

The results were satisfied, since the ML model had to classified among 33 classes, where the number 33 is not considered small for such type of computations.

After that, the prediction part took place, and 2 choices of loading data were used. The first was to use auxiliary *Pillow* library, and load the images one each time (and after that the part of the provision of the trained model took part). The second choice was via the use of *ImageDataGenerator* and feed the ML model using batches (then the classification took place). By using generators there was serious acceleration on the prediction period. It should be noted that Google Coral TPU did not support *Tensorflow* package. Instead *Tensorflow* Lite was used. All the results of all the SBCs are analyzed to the following part.

2.5.7 Analysis of the various experiments with the use of SBCs

The following outputs were measured on the four different Single Board Computers by using the built ML model and use it in the inference:

1. current, measured in mAmperes
2. voltage, measured in Volts
3. power consumption, measured in mWatts
4. CPU usage, measured in percentage (%)
5. memory swap usage, measured in percentage (%)
6. temperature, measured in ^{o}C

In order to process better the image classification in the four different IoT devices, the python library *Pillow* was used with the aim to supply each time one image, thus, saving RAM memory. However, this tactic makes the process of inference more time consuming. Also, it refers to the process of making use of supervised trained algorithm the ML model in order to produce predictions. All the measurements for the CPU usage, RAM memory, Swap memory, and for (the internal) temperature of each device were realized via the python library *psutil.* All the previous

measurements and the data were provided in a format that could be edited or stored, in a .csv file. For measurements related to power consumption, and current draw, a special measurement device was used by the researchers of the Diffused Intelligence Lab. This device was capable of datalogging voltage values, current values and power consumption values with frequency equals to 10 seconds. The frequency of the measurements can be easily adjusted to the one that the user desires. By monitoring these parameters, someone can have an overall picture of the power consumption, something very essential since the experiments are taking place to IoT devices with many constrained resources, such as: power, CPU, RAM, bandwidth, etc.

In **Figure 27**, it is obvious that the more powerful device, can also exploit this characteristic. Google Coral TPU that makes use of the ML accelerator can succeed in completing the inference part in the same time duration as the Raspberry Pi 4B, where the latter encompasses more RAM, 4 GB against 1 GB. Jetson Nano, at the early stages of the operation needs serious amount of power, but then it falls to more normal amount of power, an indication that it loads dynamically the tensorflow library at the start, so it needs more CPU resources. Another issue is the fact that classification (prediction) operation was implemented with the same dataset in all devices, which has consequences as using 20% of the starting number in every category and of course the output value was similar.

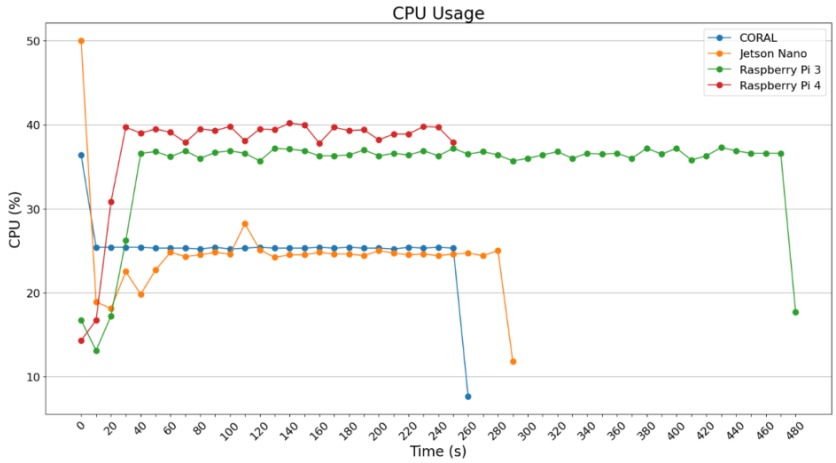


Figure 27 CPU usage using the Pillow library.

The diagrams in **Figure 27** show that Raspberry Pi 4B finished the task in around 244 seconds, very close was the Google Coral with 253 seconds. Jetson Nano finished its task at 275 seconds and Raspberry Pi 3B+ finished its task in 473 seconds. The proposed model includes a mean accuracy of more than 90%, but two classes, Tomato Septoria Leaf Spot and Tomato Late Blight, hit score of 50%.

As depicted in **Figure 27** NVIDIA Jetson Nano makes use of about the ½ CPU power (25%) in comparison to the two Raspberry (Raspberry Pi 3 (38%) and Raspberry Pi 4 (40%)). This is a sign that NVIDIA Jetson Nano uses basically its GPU power in order to accomplish the various commands that was been given through the python code, with the corresponding ML model. On the other hand, the two Raspberry Pi use as a main part their CPU in order to execute the python code, and this is the reason why they use more CPU resources in contrast to the NVIDIA Jetson Nano. Another clue, coming from the **Figure 27** is the fact that Google Coral Dev TPU

demonstrates the same behaviour as the NVIDIA Jetson Nano, meaning that it makes use less CPU power. But the latter occurs for a different reason. Google Coral TPU exploits its ASIC module so that it can manage and execute the various python ML python's commands that are linked to the tensors. The concept is that Google Coral can improve the parts of the code which are related to tensors, with a serious number of tensors, since the python script uses Tensorflow Lite package, so there is no need to take advantage of its CPU power. This is the main reason for using only 25% of its CPU instead of 38% and 40% used from the Raspberry Pi 3 and 4.

As someone can observe from the "Related Work" section, the researchers in [50] reach accuracy of 93,33%. The researchers in [51] reach the following accuracies: for the disease "Bacterial Blight": 85,89%, for "Alternaria" 84,61%, for "Cerespora" 82,97%, for "Grey Mildew" 83,78%, for "Fusarium Wilt" 82,35% and for "Healthy leaf" 80%. In the research work [52] the researchers state that they hit the following accuracies:

i) for ReLU, 7x7, testing accuracy 95,7 %, in 51 minutes.
ii) L-ReU, 7x7 testing accuracy 97,3%, in 53 minutes.
iii) L-ReLU, 11x11, testing accuracy 98,0%, in 54 minutes.

As a last comment in [53], the authors claim that they reach 97% accuracy. So, from the above accuracies, the mean score is about 89,17% which rationalize the current outputs and the claim the "about 90%" is an accepted score-threshold.

In **Figure 28** someone can see that the 2 Raspberry Pi use more their CPU power for each computation, while the NVIDIA Jetson Nano and Google Coral Dev TPU uses less CPU power due to the fact that they use more their accelerators, and more precisely the CUDA cores for the Jetson and the ASIC module for the Google Coral.

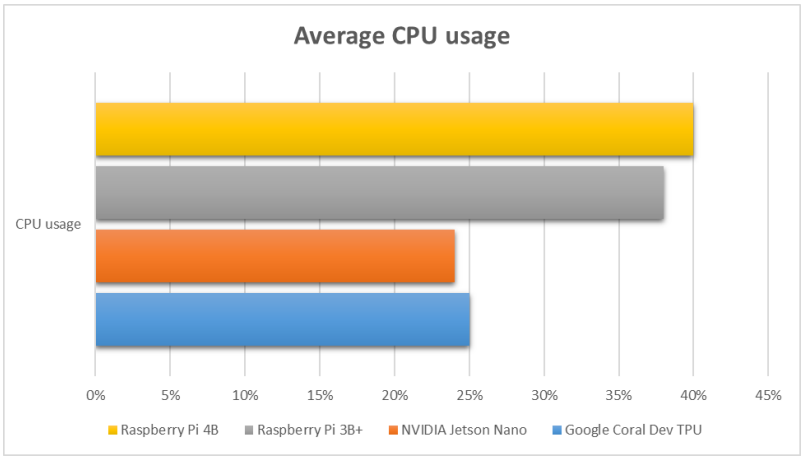


Figure 28 CPU usage of each SBC, demonstrated in percentage.

Observing the images of the 2 categories mentioned previously, Tomato Septoria Leaf Spot and Tomato Late Blight, someone can understand that the errors which occurred, are the outcome of the fact that the images are similar and the extraction of different characteristics was not easy, thus, the accuracy is 79% which was less than 90% (mean accuracy). Data augmentation was used: however, these kinds of problems are not easily excelled, because of the small number of images used during the training period, moreover when there are 33 classes for classification without connecting similar categories.

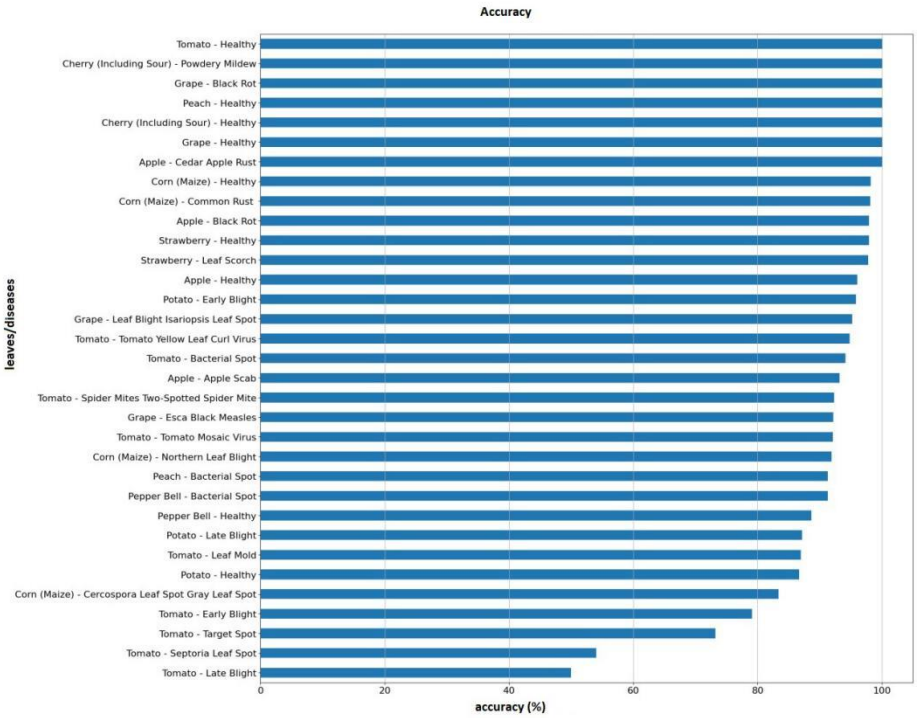


Figure 29 Accuracy for each class.

As far as the issue of identical images is concerned, and the considering problems that caused, because the output of the discrete characteristics is difficult, one solution could be the usage of more images per class. The current work used 33 classes, making it a serious number of classes for the scope of our experiments and the linked hardware used, however the images exploited in every class was not so big, and as a result they cause problems. To be more precise, the Tomato Early Blight class contained similar images, which is the reason for reaching 79% accuracy, significantly lower than the consent threshold of 90%, as claimed in previous section, for the current work. In **Figure 29**, the differences among the accuracies achieved for each class can be observed.

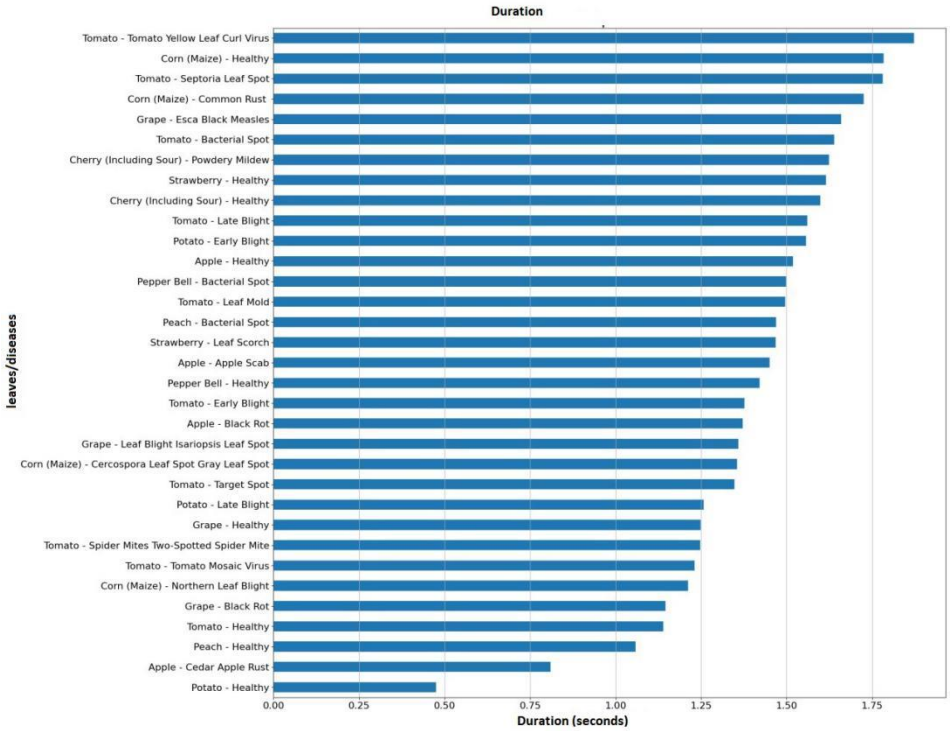


Figure 30 Inference duration for every class via the use of Google Colab.

The diagram presented in **Figure 30** shows the times needed for processing every class by using Google Colab. Comparing Raspberry Pi 4B, Jetson Nano and Google Coral with Google Colab, the time needed is 5-times more and on Raspberry Pi 3B+ the time is 10-times more than the time needed is Colab, which is also depicted in **Figure 30**, **Figure 31**. The small-time differences for each class are the outcome of the size of images, since the accurate same number to all classes was not available but a difference of +/- 5 images for each class. The result is depicted so that someone can compare the tremendous capabilities of the Colab VM Cloud against the constraint resources SBCs. In **Figure 31** and **Figure 32** is depicted the RAM usage in MBytes and in percentage.

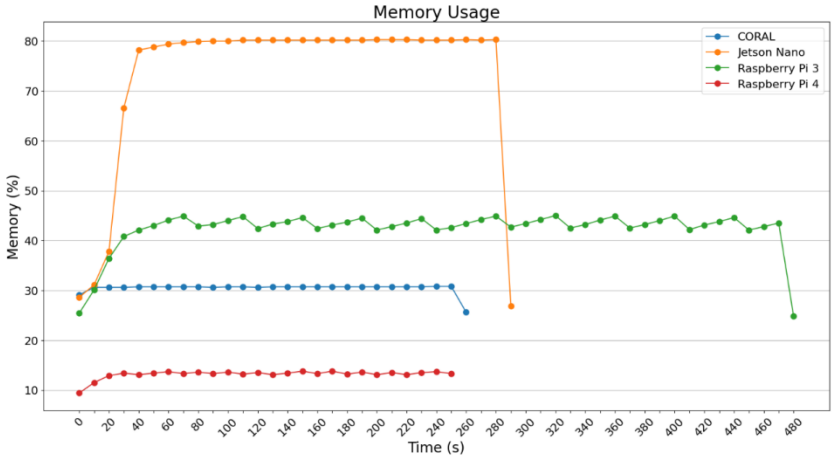


Figure 31 Used memory with Pillow, depicted in percentage.

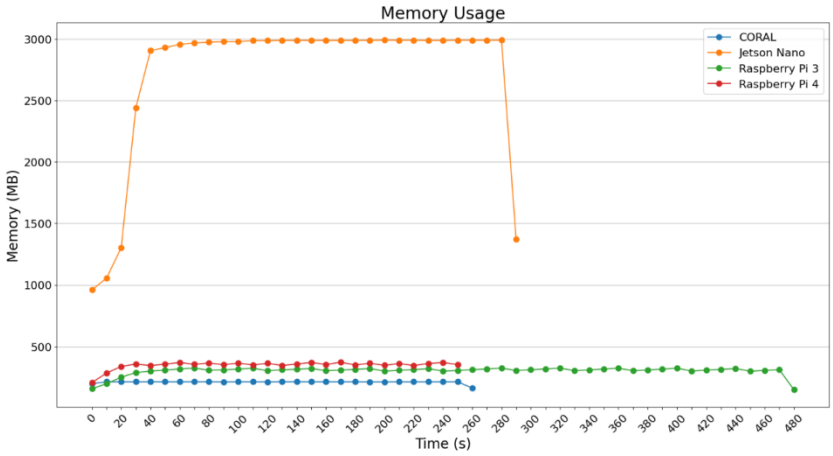


Figure 32 Memory usage depicted in MBytes when Pillow is chosen to operate.

Jetson Nano needs a lot of RAM memory, while the other three devices need no more than 500 MB. More specifically, Raspberry Pi 3B+ needs lower than 50% of its total RAM memory, Google Coral does not need more than 30% of its total memory, and finally Raspberry Pi 4B does not use more than 15% of its available RAM. In **Figure 32**, **Figure 34**, it is more than clear that Jetson Nano uses GPU in most of its processing time for compute the various operations, which are energy hungry.

In **Figure 33** someone can see that NVIDIA Jetson Nano makes use more of its available RAM, where 4 GB RAM is its total memory, then follows the Raspberry Pi 3B+ (with 1 GB total memory) and then Raspberry Pi 4B (with 4 GB total memory). All the above in real numbers stand for the following: NVIDIA Jetson Nano uses 3,2 GB RAM, Raspberry Pi 3B+ uses 450 MB RAM, Google Coral uses 300 MB (Mega Byte) RAM and Raspberry Pi 4B uses 600 MB RAM.

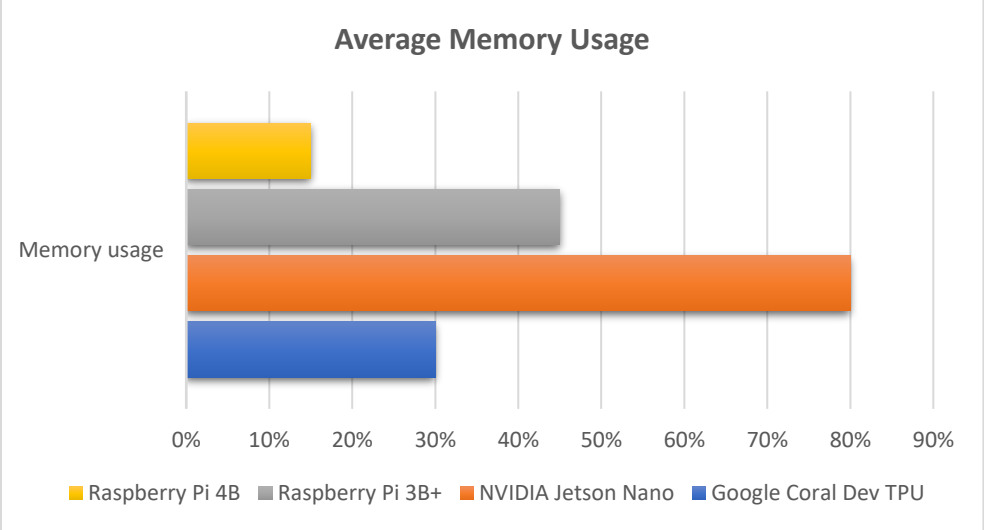


Figure 33 Memory usage of each SBC, depicted in percentage in contrast to its total available memory.

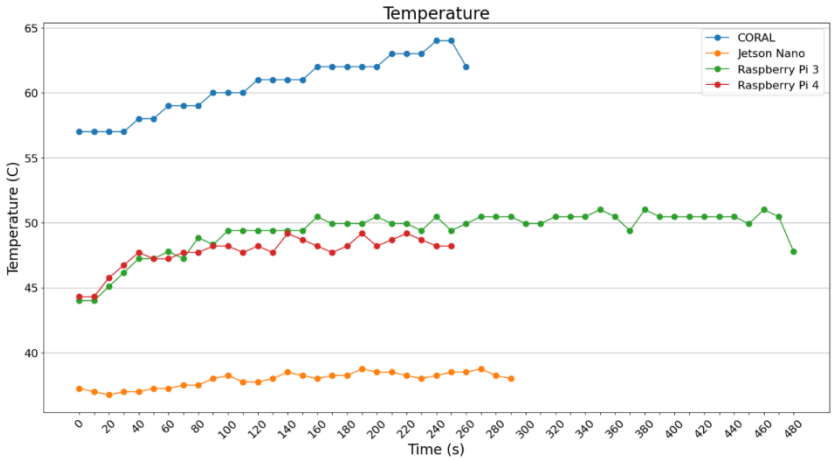


Figure 34 Internal temperature of the SBCs when Pillow is used.

In **Figure 35** someone can see that NVIDIA has the least temperature while working against all the 4 SBCs. The latter occurs since it uses a large heatsink and a bigger fan than the other 3 SBCs. Google Coral Dev TPU, cannot manage the increased temperature of around 60 °C and reaches the most temperature of all the 4 SBCs. The two Raspberry Pi reaches around 48 °C and 50 °C and although the fact that both SBCs contain fans and heatsinks operating continuously, they do not seem so effective in reducing the heat.

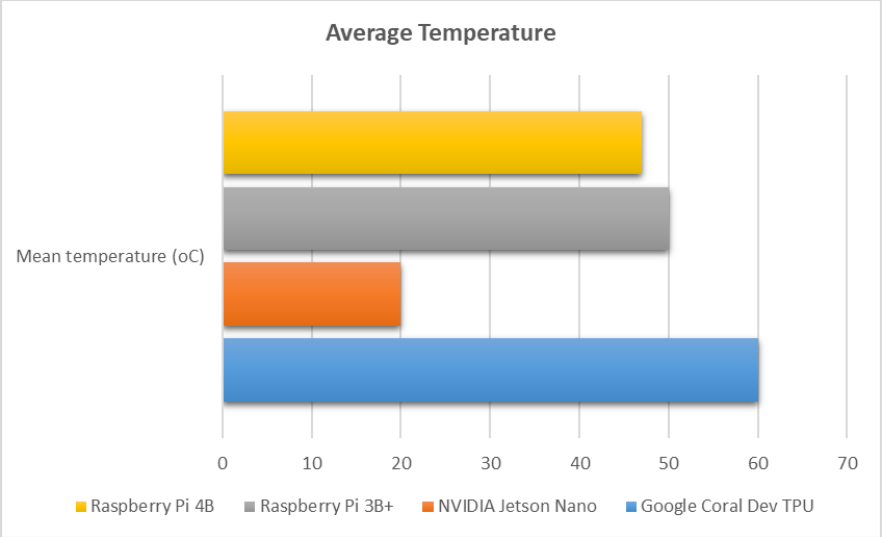


Figure 35 Memory usage of SBC, depicted in percentage in contrast to its total available memory.

Concerning the temperature measurements, Jetson Nano maintained low temperatures while operating, maybe because of the very good cooling system it encompasses: a large heat-sink and a fan. However, Google Coral measured with high temperature, which looks like it controlled it because the embedded fan did not operate of the time, but was triggered and paused when needed. **Figure 36** depicts the temperatures for all the SBCs. All of them are supplied with 5.1 Volt DC, although in **Figure 36** only the current draw is presented. The power supplied to each SBC is given by the following formula:

$$P(power(mW)) = 5.1(Volt) \cdot I(Current(mA))$$

As someone can observe all the SBCs apart from the Google Coral need 1000 mA current with a few numbers of peaks more than 1350 mA while the other values maintained above 800 mA. Google Coral because of its architecture manages values without a lot of variances when drawing current, and as someone can observe it is below the related current measurements of the rest of the devices.

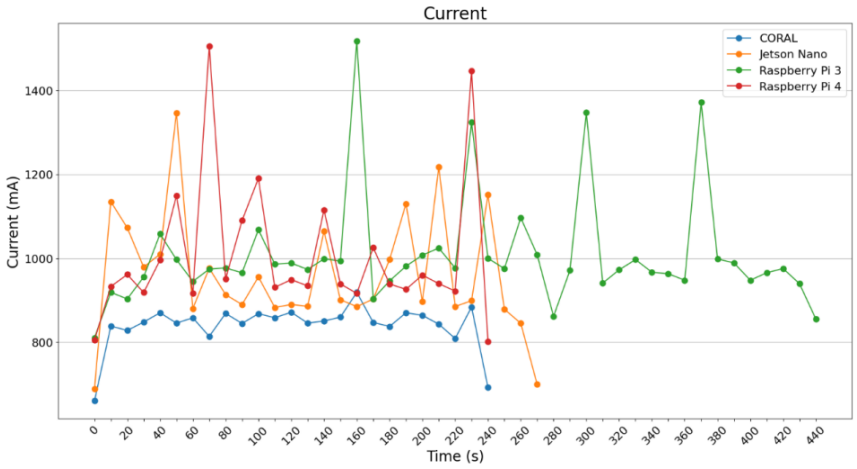


Figure 36 The current that each device needs when using *Pillow*.

In **Figure 37**, it is observed that NVIDIA Jetson Nano consumes the most energy of all the 4 SBCs, and this is something reasonable, since it is well-known that GPUs need a lot of power to operate. The latter is one reason that scientific community uses also modules such as ASICs or more generally (FPGAs – Field Programmable Gate Arrays[34]), that can speed up the process using low power.

[34] https://inaccel.com/studio/

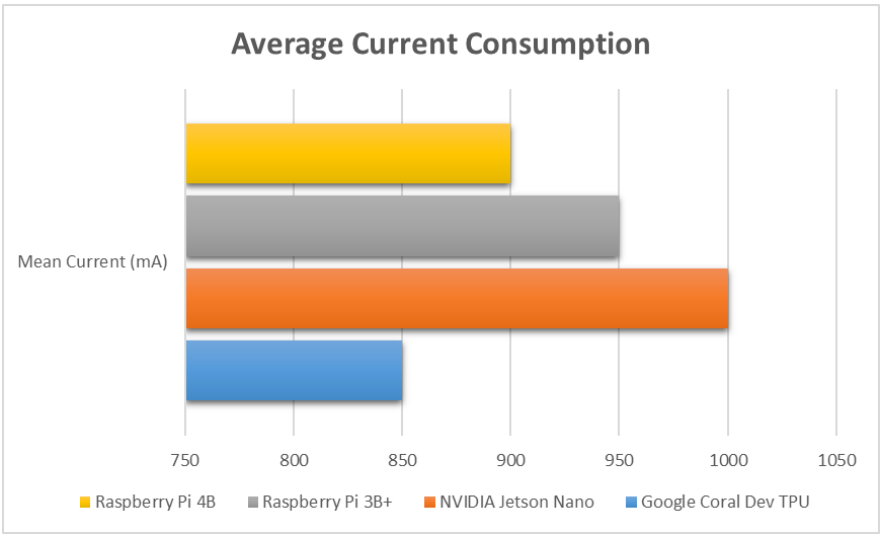


Figure 37 Average current consumption of each SBC, presented in milli-Amperes.

Using generators has also consequences to the prediction period, apart from the training period. The images are loaded per batch in the model. They are processed in groups, such as: 2, 4, 8, 16, 32. For the implementation of the various experiments, presented in the current chapter, only Raspberry Pi 3 and Raspberry Pi 4 and not Google Coral TPU where used, because the latter does not provide support for the common tensorflow, so *ImageDataGenerator* could not be used, because it is part of Keras library. Keras is an open-source library that supply python interface for the ML model. It is an interface for tensorflow package. Apart from Google Coral TPU, nor Jetson Nano was exploited due to the fact that the OS it encompasses did not support that operation. In the experiments, batch_size equals to 2, 4, 8, 16 images as used, and batch_size = 32 only for the Raspberry Pi 4B was used, showing better results in the execution time. In the following Figures, one can observe decrease in execution time and more need for computational resources, while increasing the batch size. In **Figure 38**, the results for *batch_size* = 2 are depicted.

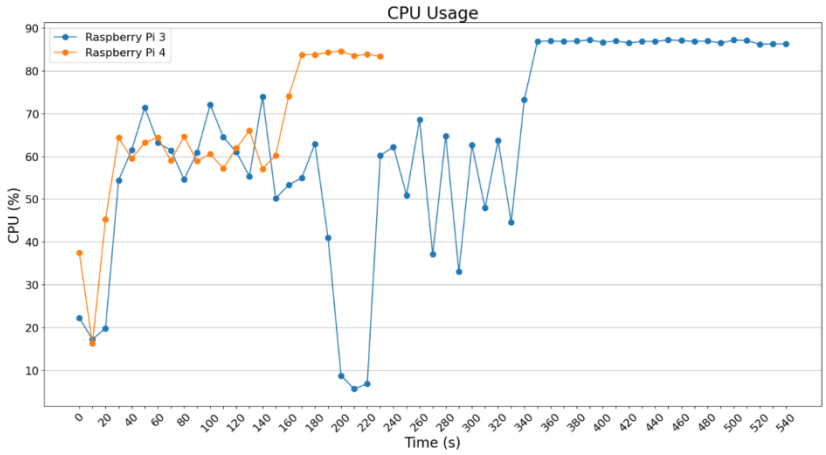


Figure 38 CPU usage, with ImageDataGenerator and batch_size = 2.

For batch_size = 2, nothing notable was noticed, except for the fact that Raspberry Pi 4B can operate faster. Below, the results for batch_size = 4 are presented.

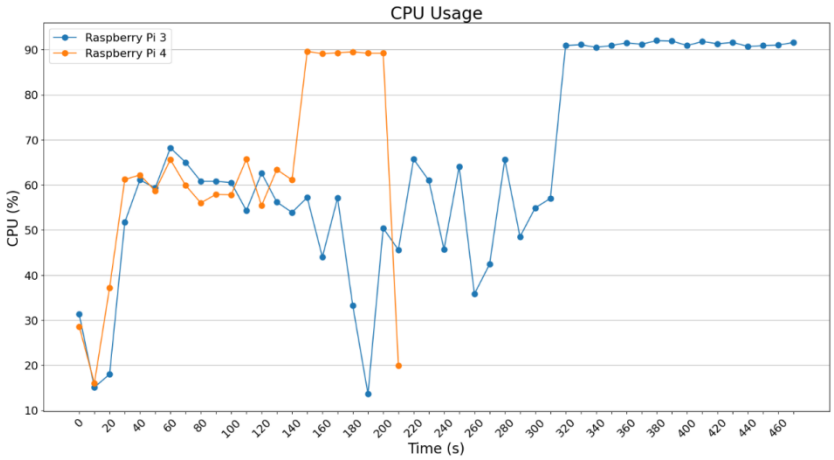


Figure 39 CPU usage, with ImageDataGenerator and batch_size = 4.

In **Figure 39** what someone can observe is the fact that Raspberry Pi 3B+ showed improved execution time in respect to **Figure 38**. Also, Raspberry Pi 4B bettered its time for 80 seconds, something quite notable in respect to the total amount of time. Below, the diagrams for batch_size = 8 and batch_size = 16 are presented.

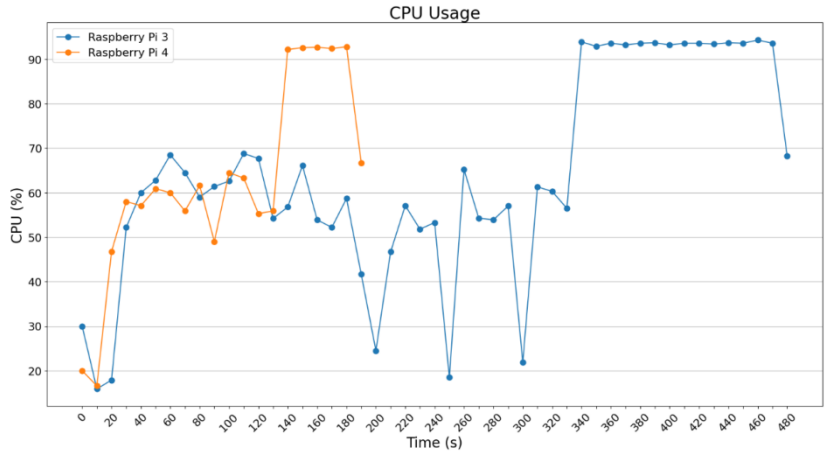


Figure 40 CPU usage, with ImageDataGenerator and batch_size = 8.

In **Figure 40**, which presents results for batch_size = 8, the small improvement is notable, but the whole picture stays the same as in **Figure 39**. In **Figure 41**, someone can see a serious improvement in the time needed for the execution for both Raspberry. The reason is that the similar rise in the number of images results in a mitigation of the recursions, which is reasonable since the algorithm realizes less iterations on the execution part.

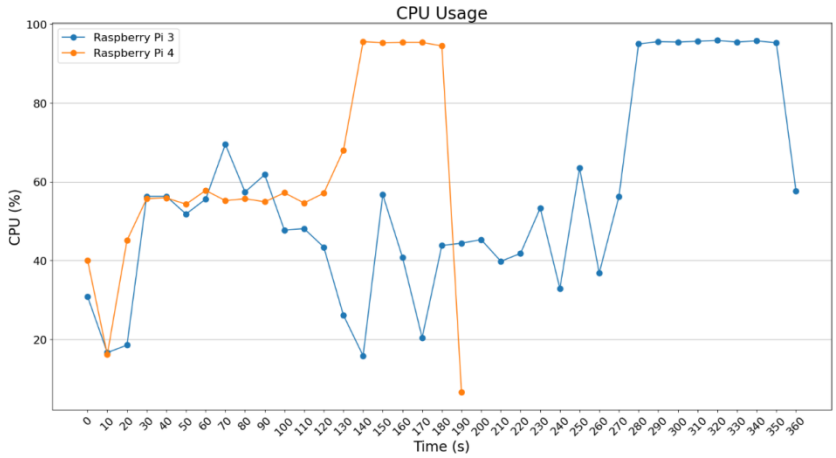


Figure 41 CPU usage, with ImageDataGenerator and batch_size = 16.

Figure 42, **Figure 43**, **Figure 44**, **Figure 45** depict the RAM behaviour in percentage for both Raspberry Pi 3B+ and Raspberry Pi 4B, whereas **Figure 46**, **Figure 47**, **Figure 48**, **Figure 49** present the memory behaviour in MBytes.

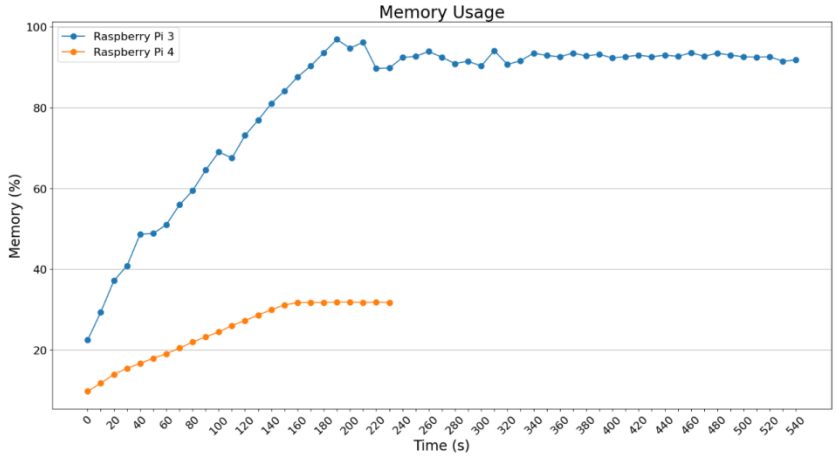


Figure 42 Memory usage in percentage, with ImageDataGenerator and batch_size = 2.

Figure 42 and **Figure 43**, indicate that RAM usage for the Raspberry Pi 3B+is maintained at high level, around 90%. The initial idea was that the Raspberry Pi 3B+ could not finish the operations because of the extra load. However, in the following figures it will be obvious that apart from borderline usage of RAM, the more memory that is necessary, it was enabled. The Raspberry Pi 4B did not have any issues as happened with the Raspberry Pi 3B+. The former achieved RAM usage of 30% - 40% in order to realize and accelerate the computations.

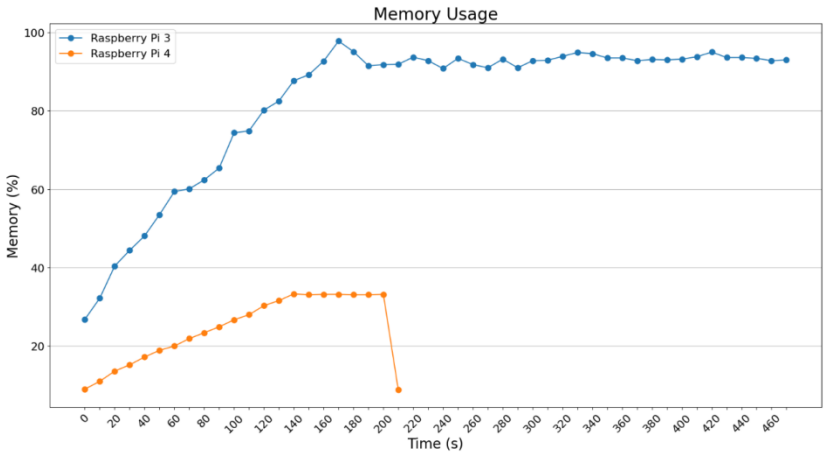


Figure 43 Memory usage in percentage, with ImageDataGenerator and batch_size = 4.

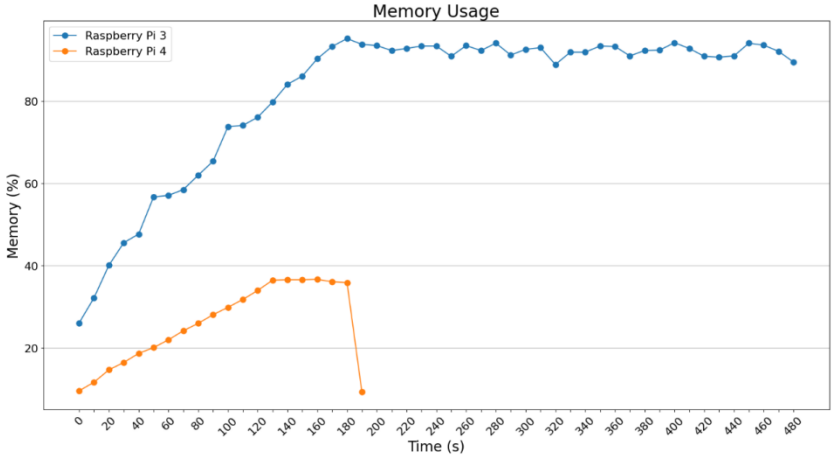


Figure 44 Memory usage in percentage, with ImageDataGenerator and batch_size = 8.

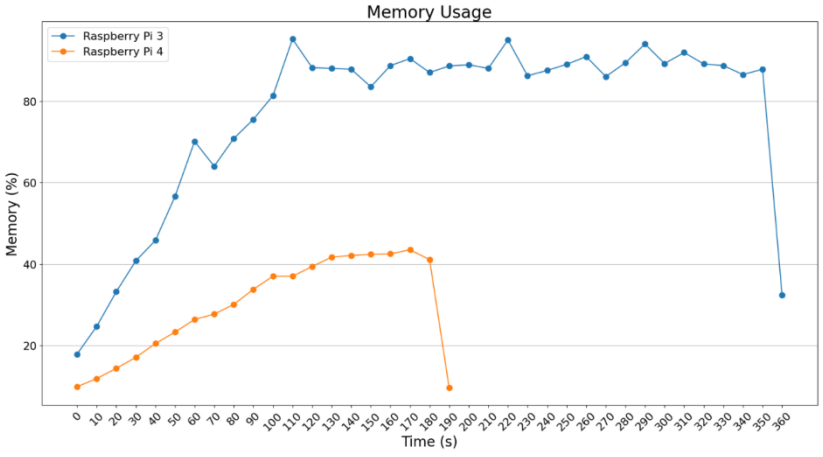


Figure 45 Memory usage in percentage, with ImageDataGenerator and batch_size = 16.

Below, are presented the results on memory usage in MBytes.

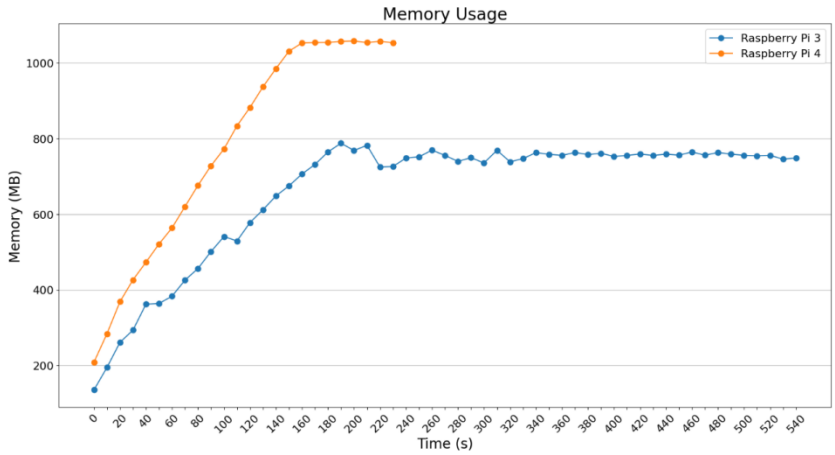


Figure 46 Memory usage in MBytes, with ImageDataGenerator and batch_size = 2.

Figure 46 and **Figure 47** depict the RAM usage measured in MBytes for batch_size = 2 and batch_size = 4. Raspberry Pi 3B+ needs about 800 MBytes (out of 875 MB total available). Raspberry Pi 4B needs more MBytes due to the fact that the total available is around 4 GB. Thus, the Raspberry Pi 4B is a significant powerful IoT SBC, something easily seen from the operation time of process.

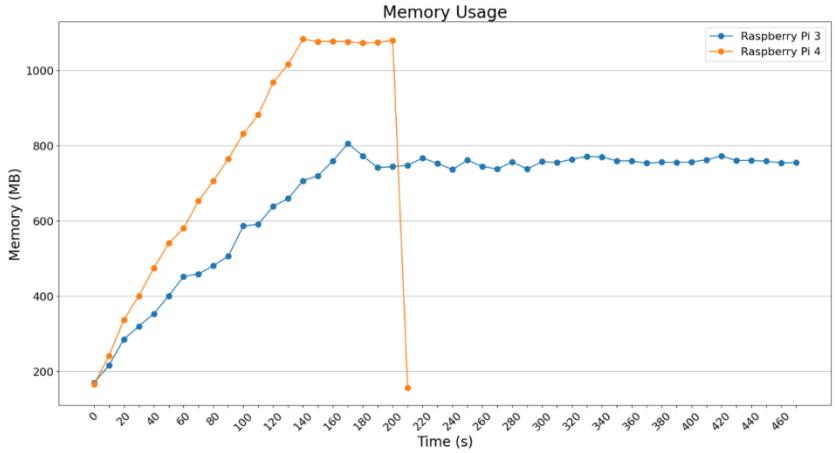


Figure 47 Memory usage in MBytes, with ImageDataGenerator and batch_size = 4.

Figure 48 and **Figure 49** demonstrate the usage of memory in MBytes in two very demanding tasks, related to batch_size = 8 and batch_size = 16.

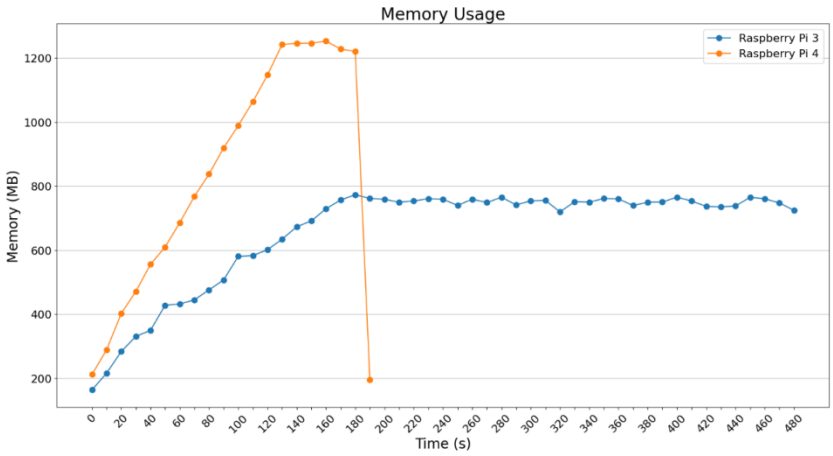


Figure 48 Memory usage in MBytes, with ImageDataGenerator and batch_size = 8.

The feedback that someone can get by observing **Figure 48** and **Figure 49**, is the fact that Raspberry Pi 4B operate better than the previously mentioned results, considering that it is fed with 8 and 16 images, needing 1250 MBytes and 1500 MBytes RAM. Raspberry Pi 3B+ gives the impression to be stressed because of the more load commended to it.

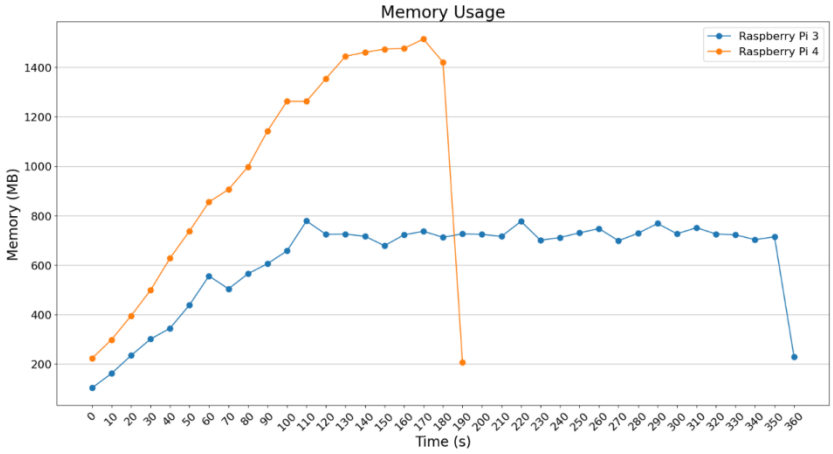


Figure 49 Memory usage in MBytes, with ImageDataGenerator and batch_size = 16.

As it is clear from the **Figure 50**, **Figure 51**, nothing special is observed. Raspberry Pi 4B seems to have increased temperature in its CPU, when operating. For batch_size = 2 and batch_size = 4 Raspberry Pi 3B+ was around 49°C, while Raspberry Pi 4B measured between 45°C to 55°C.

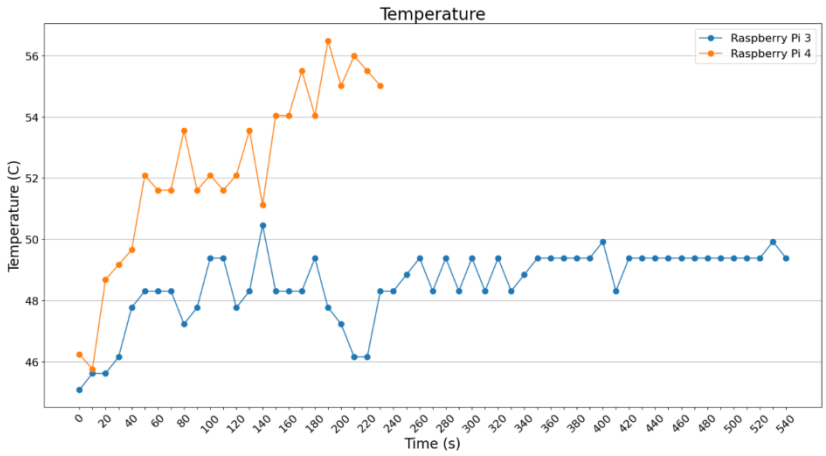


Figure 50 Temperature of both devices with ImageDataGenerator and batch size = 2.

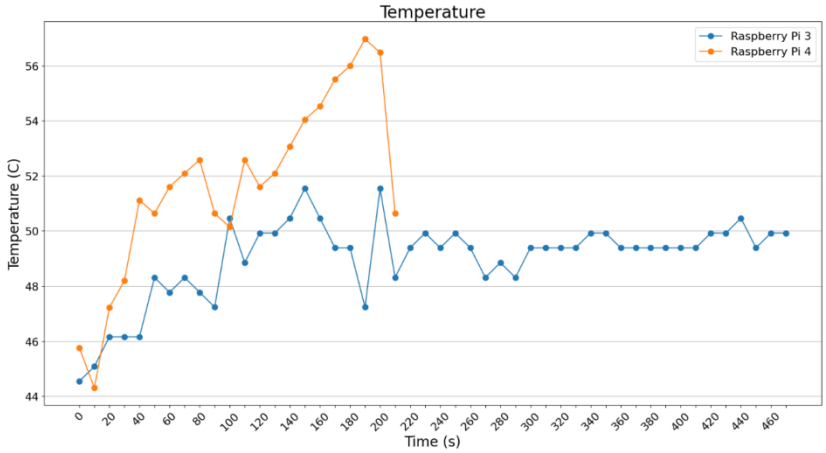


Figure 51 Temperature of both devices with ImageDataGenerator and batch size = 4.

In **Figure 52** and **Figure 53** the results for batch_size = 8 and batch_size = 16 are displayed. It is notable that Raspberry Pi 3B+ stays at 49°C including few spikes in the end, as a matter of stressing both RAM and CPU usage. For Raspberry Pi 4B, it is worth noticing that it begins normal but it ends a bit stressed.

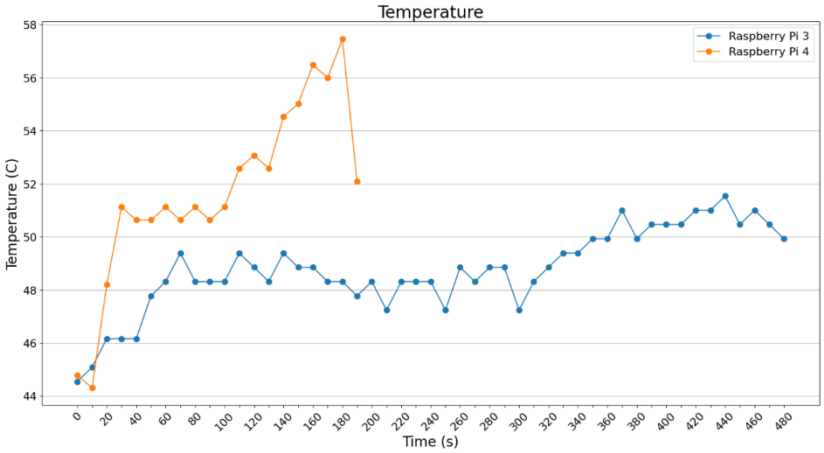


Figure 52 Temperature of both devices with ImageDataGenerator and batch size = 8.

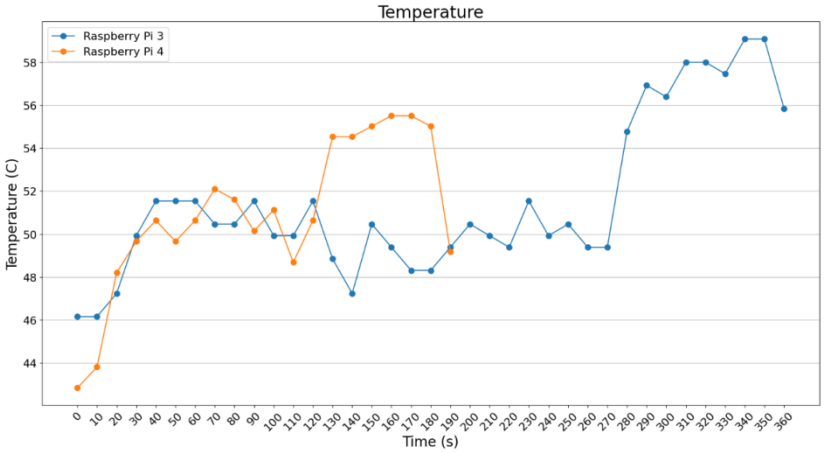


Figure 53 Temperature of both devices with ImageDataGenerator and batch size = 16.

Concerning the current per batch size, the idea that comes out from the different experiments is that the bigger the batch size, the more stressed the SBC, which has consequences to the power consumption. At the beginning the average value of the current is around 1050 mA for batch_size = 2 and rises to around 1200 mA for batch_size = 16. These numbers are for Raspberry Pi 4B. For Raspberry Pi 3B+ the numbers are even higher. In **Figure 54** it is observed many spikes for Raspberry Pi 4B but fewer for Raspberry Pi 3B+.

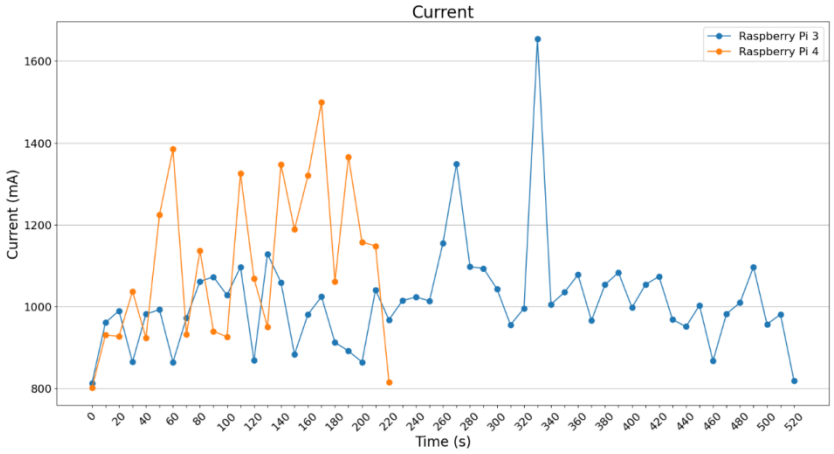


Figure 54 Current draw in mA of both devices with ImageDataGenerator and batch size = 2.

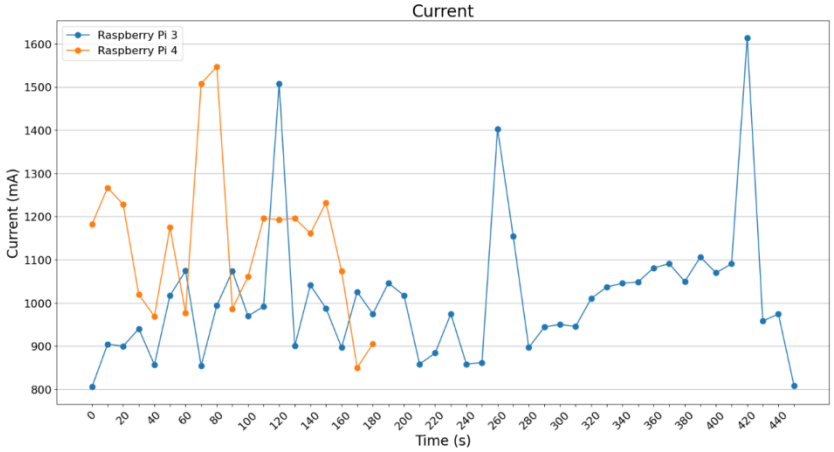


Figure 55 Current draw in mA of both devices with ImageDataGenerator and batch size = 4.

When the batch_size = 4, it is more than obvious that there is a tiny rise in current draw from the Raspberry Pi 4B at around 1200 mA. Raspberry Pi 3B+ begins to stressed where the spikes hit 1500 mA. It is clear from the **Figure 56**, that for batch_size = 8, there is a small rise in the current for the two Raspberry Pi.

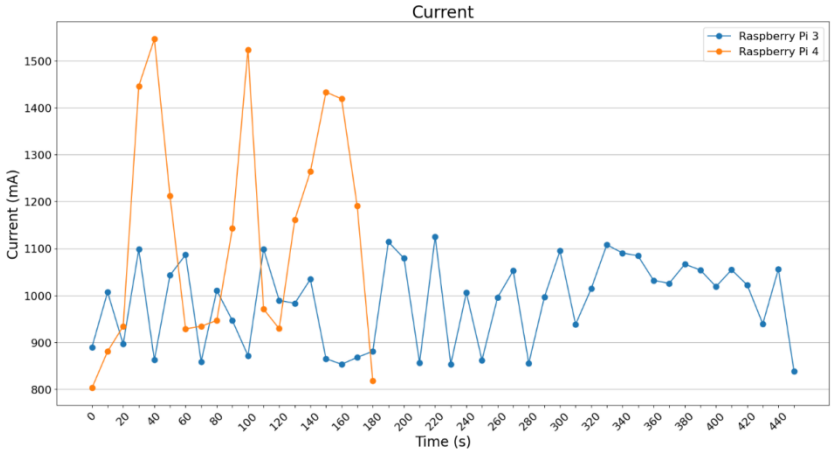


Figure 56 Current draw in mA of both devices with ImageDataGenerator and batch size = 8.

In **Figure 57** a serious difference between the two can someone observe. When the batch_size = 16, a tiny rise in current draw, in respect to batch_size = 8. Raspberry Pi 3B+'s current is more than 1700 mA for more than the half of the time, while Raspberry Pi 4B draws around 1200 mA.

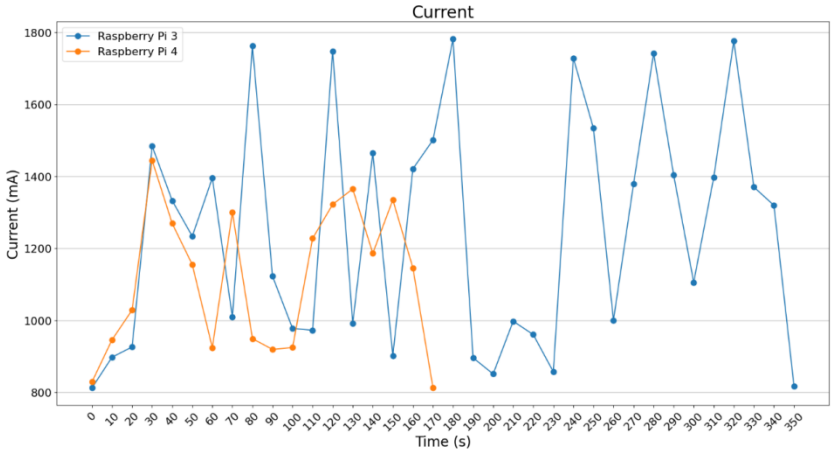


Figure 57 Current draw in mA of both devices with ImageDataGenerator and batch size = 16.

In **Figure 58** it is displayed the way the Raspberry Pi 3B+ and Raspberry Pi 4B make use of the batch_size in alternative sizes and the way every device is compared to its own for certain executions using the Pillow scheme. For instance, Raspberry Pi 3B+ outputs the measurements shown in **Figure 58**. It is crystal clear that if someone needs to earn time in the execution part, more resources are needed, especially for batch_size = 16.

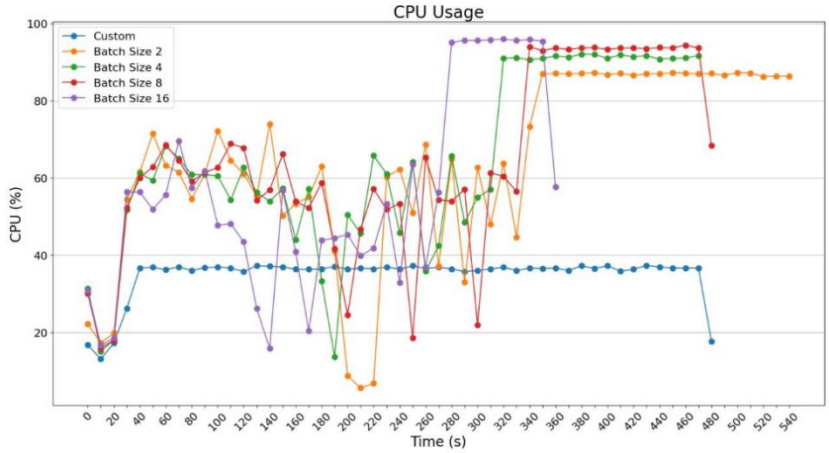


Figure 58 The CPU usage of Raspberry Pi 3B+.

In **Figure 58** and **Figure 59**, the following can be seen: the CPU behaviour and the RAM memory behaviour as far as the Raspberry Pi 3B+ is concerned, for various executions. Someone can identify that the usage of the custom choice, where the images are fed one each time, without the use of extended computational resources in order to achieve the desired output. Via the use of *ImageDataGenerator* there is an appropriate use of images per batch, thus the processing of every group decreased in a borderline in execution time.

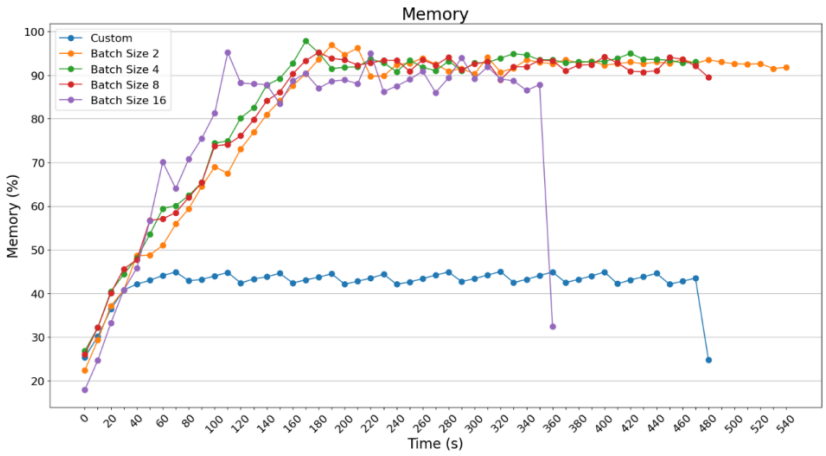


Figure 59 Percentage of RAM memory usage while using Raspberry Pi 3B+.

In **Figure 60** someone can see that the values for memory usage in MBytes have similar appearance to the diagram showing the percentage using memory RAM.

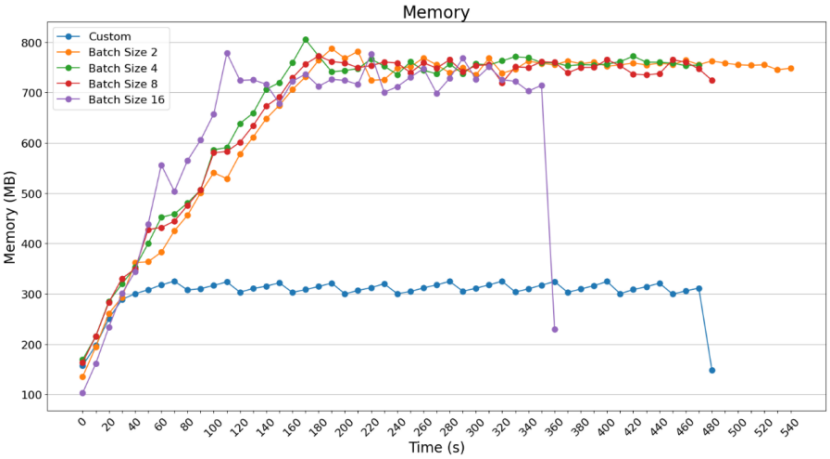


Figure 60 SIze (in MBytes) of RAM memory usage while using Raspberry Pi 3B+.

Figure 61 presents the additional need as an output of bigger size group when loading images, due to the more need for memory. Raspberry Pi 3B+ needs the double size, thus, it uses 750 MB Swap, in order to finish the computations in a demanding task.

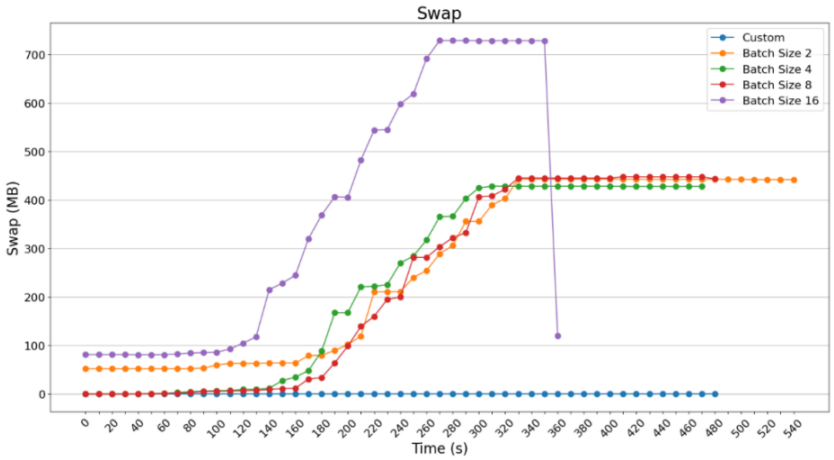


Figure 61 SIze (in MBytes) of Swap memory usage while using Raspberry Pi 3B+.

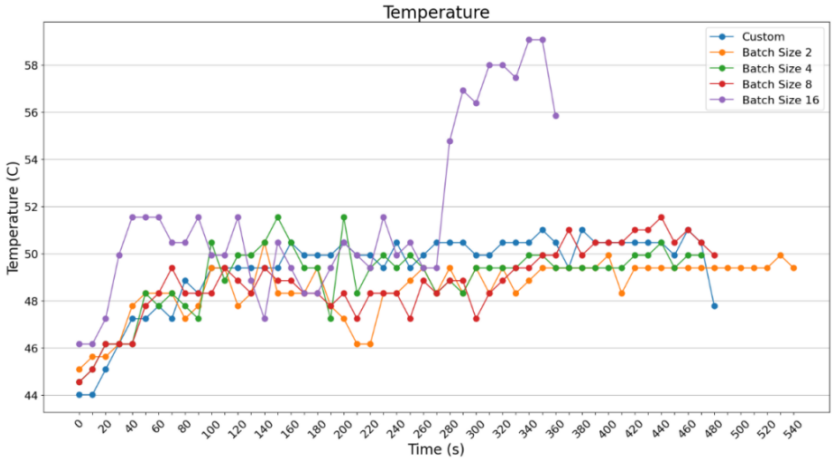


Figure 62 Temperature behaviour of Raspberry Pi 3B+.

As it is observed there a significant difference that occurs between the execution for batch_size = 16 and the rest batch sizes. As far as the temperature is concerned in **Figure 62** there was no strong outcomes, apart from the fact that for batch_size = 16, the time looks like to be separated from the other batch sizes. Concerning Raspberry Pi 4B with batch_size = 32, it succeeds in giving better results, thus it seems to exploit its hardware better.

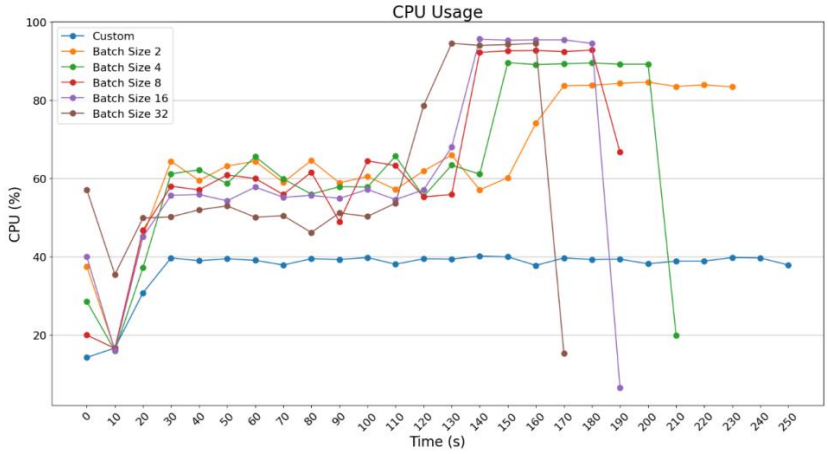


Figure 63 Raspberry Pi 4B results, concerning CPU usage.

Concerning the usage of computational resources, there is no serious difference to that various batch sizes, although *ImageDataGenerator* was used. One thing that distinguishes is the comparison between the custom built and the rest.

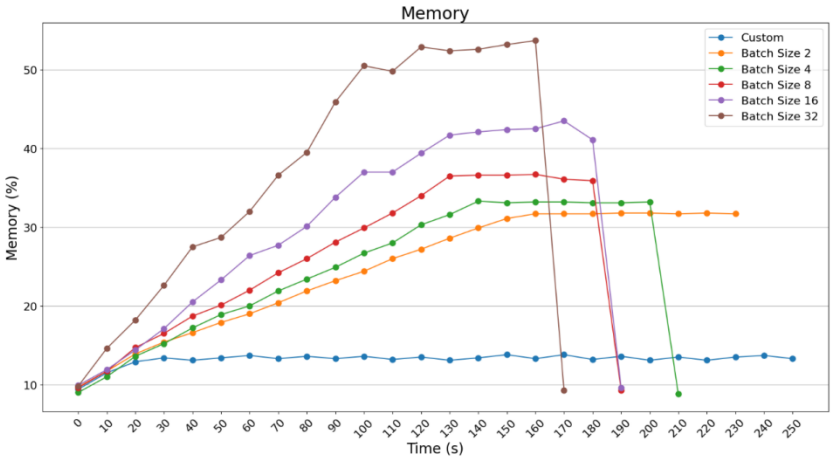


Figure 64 Raspberry Pi 4B results, concerning RAM memory usage, depicted in percentage.

In **Figure 65** it is obvious the acceleration that its duration took about 20 seconds, by using batch_size = 32. To achieve this number, there was a doubling in the number of images for every group, that is why the percentage of RAM memory exceeds 50%. Strain forward was the RAM measured in MBytes where it went over 1800 MBytes, while in the previous results it did not exceed 1500 MBytes.

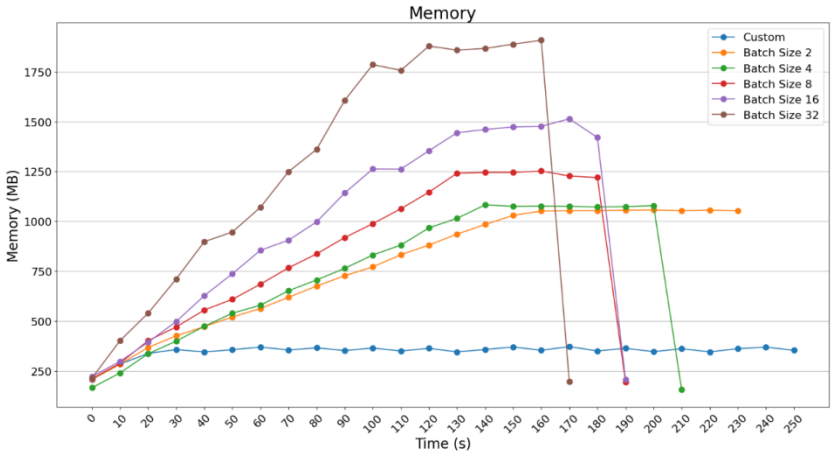


Figure 65 Raspberry Pi 4B results, concerning RAM memory usage, depicted in MBytes.

In **Figure 66**, it is obvious that the temperature results do not indicate a significant difference for different batch sizes. The Raspberry Pi 4B keeps the temperature at about 52°C average value.

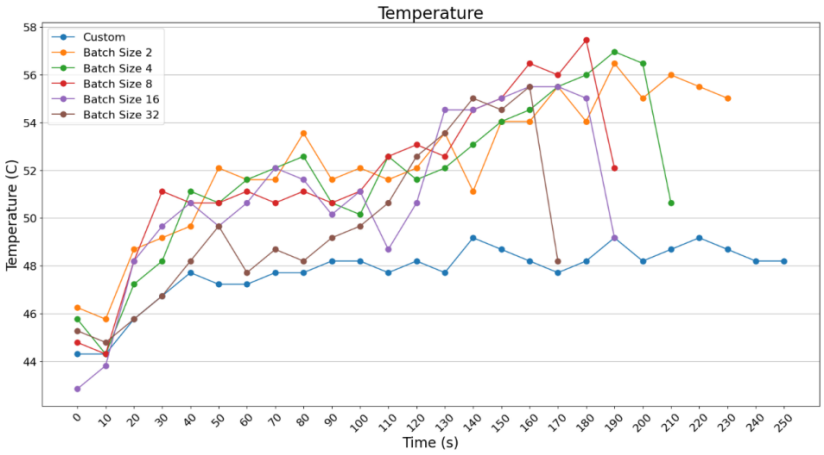


Figure 66 Raspberry Pi 4B temperature results.

Although an improvement in the execution time duration took place with batch_size = 32, and the appropriate rise on RAM memory resources, very interesting is the fact that Raspberry Pi 4B can quite easily realize the prediction with *Pillow* that represents the customized solution for feeding images one-by-one via ImageDataGenerator, apart the fact that it needs 80 seconds more.

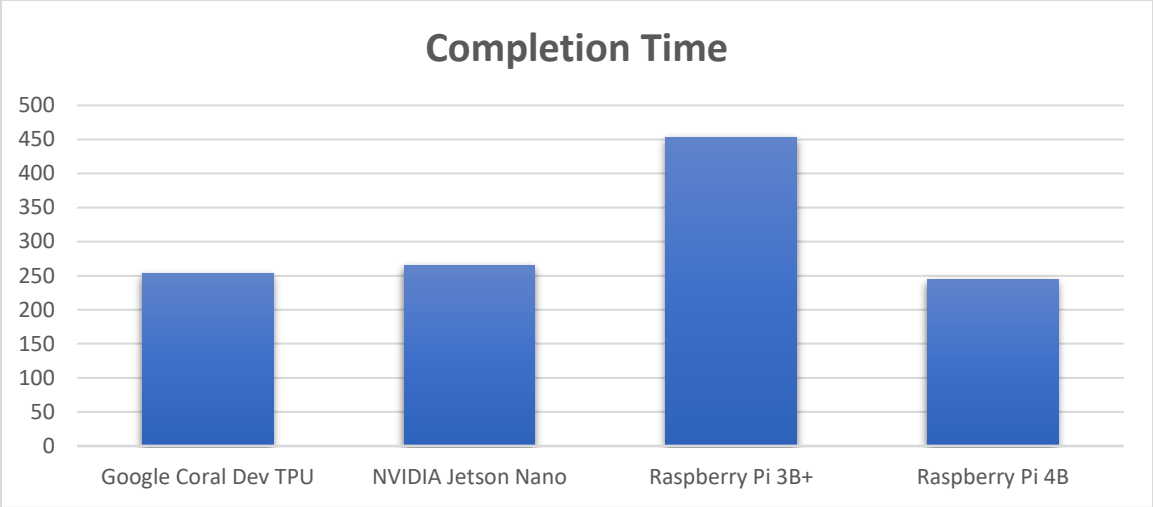


Figure 67 Completion time of every SBC on the inference part.

As it is more than obvious from the graph, the Raspberry Pi 4B is the fastest of all SBCs it the time it needs to complete the task, and as it was referred to a previous section, it takes about 244 seconds to complete the task, better known as the inference part of the ML model. The slowest SBC is the Raspberry Pi 3B+, which needs about 453 seconds in order to complete the task. Raspberry Pi 4B uses 4GB RAM and more capable processor: Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz in comparison to Raspberry's Pi 3B+ CPU: Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz, as it depicted in **Figure 67**. The latter also makes use of 1GB RAM, seriously limited in contrast to the 4GB RAM of Raspberry Pi 4B. The perception indicates that NVIDIA Jetson Nano must have been the fastest of all the 4 experimented SBCs, but it is not. As it is crystal clear from the graphs, TPU and GPU use less CPU power than the Raspberry Pi which are CPU-based.

SBC	Mean Current (mA)	Voltage (Volts)	Power (mW)	GFLOPs or GOPs	(GFLOPs or GOPs)/mW
Google Coral Dev TPU	850	5	4250	4000	0.9411
NVIDIA Jetson Nano	1000	5	5000	472	0.0944
Raspberry Pi 4B	900	5	4500	9.69	0.0021
Raspberry Pi 3B+	950	5	4750	5.3	0.0011

Table 7 GFLOPs stands for Giga (10^9) floating point operations per second and is implemented to Raspberry Pi 3B+, Raspberry Pi 4B, and NVIDIA Jetson Nano. Whereas GOPs stands for Giga (10^9) operations per second and is implemented to Google Coral Dev TPU.

It is clear from **Table 7**, when talking about inference part, the Google Coral Dev TPU is the most efficient with 0,9411 GOPs/mW, with the NVIDIA Jetson Nano following (0,0944 GFLOPs/mW), with Raspberry Pi 4B (0,0021 GFLOPs/mW) and the slowest is the Raspberry Pi 3B+ (0,0011 GFLOPs/mW). Although, GFLOPs and GOPs are different, we give a sense of their relation with the milli Watts. Google Coral Dev TPU is the most efficient of all the 4 compared SBCs as it is depicted in **Figure 68**.

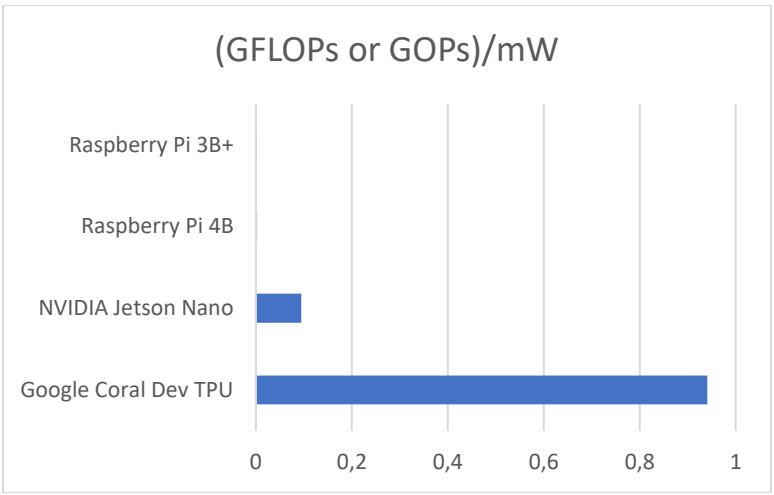


Figure 68 Giga (10^9) floating point operations or Giga operations per second per 1 milli-Watt power consumption for each SBC.

2.5.8 Limitations and Risks in the Current Work

When using SBC, it is very crucial to execute python code that exploits the various CPU/GPU/TPU constraints as well as the diminished RAM memory, and the reduced ROM capabilities. A serious issue is the fact that the initial python ML code has undergone the training phase in Google Colab, which offers powerful GPU units, and extreme parallelization in the used python scripts. It is very difficult, moreover, not advisable to train a Machine Learning model in an SBC, because it can take many hours or days. SBCs contain many limitations that make them infeasible to train there a ML model. The limitations range from CPU, RAM memory, swap memory, HDD to power consumption. Experience in the past, indicates that although training in GPU-centered SBCs need less time than CPU-centered SBCs, it is not advised at all. Another problem that was met, was the fact that Google Coral TPU cannot exploit the mainstream python Tensorflow package, but a lighter edition, the Tensorflow Lite package. The latter caused some limitations to our python code, so changes had to be made in order to be able to executed in Google Coral Dev TPU.

2.5.9 General evaluation results of the experiments

In the image processing period, the applications that were studied at the time of writing the current section (2024), are related to one type of leaf, which was divided in basic leaves' classes. In the current work, various leaves were exploited, spread in more classes rather than using one leaf, because in an image depicting only one type of crop there are different kinds of yields, and a general action should take place. There was a try to use images close to reality and not perfect images. To accomplish this, the following were addressed: light variations, and other artificial changes that depict reality, in order to make our model to be more efficient in real case prediction.

The following are the outcomes of the metrics: the GPUs need a lot of energy to execute the ML python code. They provide extreme parallelization and reduce the completion time, but the latter comes with a trade-off. Raspberry Pi 4B seems faster than the older and less powerful Raspberry Pi 3B+, when finishing a task, however both are CPU-based and in large ML models they need a lot of time because they do not use accelerators. Google Coral Dev TPU is considered fast at

finishing the job, as it basically uses less CPU power and lower energy consumption than a GPU-centered SBC, since it accelerates very specific parts of the python code. But it needs configurations to the code, moreover to the models that it can accelerate since it uses the lightweight Tensorflow Lite package.

The increased temperatures existing in an SBC are very critical when there is necessity to use them far from wall plug power supply. To give an example, if the SBC with some sensors or even a small camera needs to operate with power coming from a solar panel or a small wind generator, there are power supply limitations. So, it is very critical to mitigate the unnecessary energy, and use it to the most important elements only. Another issue when using one of the proposed devices during hot periods, such as the summer, is the fact that the hot should be dissipated using bigger fans and extra equipment such as heatsinks. The heatsinks do not consume energy, but the fan or fans need current, something that puts an extra load to our device which uses battery. So, again there is a constraint that should be taken into consideration.

Experience with Google Colab large-scale ML schemes has indicated that a GPU can provide acceleration in the training period of an ML model against a CPU. The rationale behind this, is the fact that a GPU can offer extreme parallelization, with thousands of threads operating at the same time, in parallel, while on the other side CPU works sequential and consumes more time than a GPU. The last years another solution appeared, that of the TPU, which accelerates the ML model in comparison to the CPU, depending on the number of tensors used.

2.6 Conclusions

The current Chapter introduced a modern approach in the agriculture field, targeting to help users in enhancing the management of resources and provide decision-making operations. An ML-centered image processing application was presented in order to classify plant photos across 33 classes of leave diseases reaching an accuracy of around 90%. Compared to the accuracy of the rest state-of-the-art examples, the proposed choice indicates to be more efficacious. The researchers succeed in classifying the images across many clusters, that seriously increased the difficulty of the current work: older work that reached 90% accuracy took advantage of fewer categories. Furthermore, the current chapter also presented that via the auxiliary use of generators and data augmentation, the required time for training and model realization was mitigated. This is also guaranteed for the model-based knowledge required time when taking advantage of IoT devices with limited resources (SBCs).

Furthermore, concerning image processing, the current chapter analysed a number of experiments involving IoT devices, with the goal of recording power consumption and current usage. The related metrics, have been chosen for more study, because there is occasionally the need to produce a bigger cluster using devices like the ones presented. Moreover, energy limitations and, in general, mitigated consumption and other limitations, are of uttermost significance. It has been seen that the acceleration of operations made the units increasingly energy-hungry. For this reason, it is up to the studied use-case and it depends to the developer to choose the best trade-off between presented performance and used resources.

Finally, as summed up in the Related Work Section, ML models aiming at the agricultural sector or different sectors executed on SBCs with reduced power solutions have shown several

constraints. The latter occurs because CPU, TPU and GPU are less capable than desktop PCs, something noticeable when Cloud computing resources are used. As someone can easily understand, this has serious consequences related to the completion time of the inference period as depicted in the related graphs in the sections above. Another essential factor is the fact that low-power hardware is connected with limited RAM: ranging from 1 GB to 4 GB, which inserts another limitation in the field aimed by the current chapter. As far as the hardware solutions are concerned, when the proposed approach required to operate only with energy coming from battery instead of wall plug, it is advised to supplement the battery energy source via the use of a small solar panel or a small wind generator, specifically when the NVIDIA Jetson Nano is used, which is the most "energy-hungry" of all the four SBCs analyzed in the current chapter. At this point is should be noticed that cooling solutions are advised when the end hardware devices are used during hot periods like summer, which as can someone understand insert the parameter of extract current for the cooling fans. It was depicted in the previously related chapter that when the batch_size rises, the completion time falls, but more resources are needed guiding to more stressed units.

The future research plans of the researchers contain the expansion of the image classifier so that it makes good use of UAVs transmitting images in real time over an agri area. These images will be gathered by a remote IoT device, which uses the expanded application, programmed to help real-time decision-making operations concerning the leaves' diseases. The decision making will be matched with many environmental and agricultural measurements gathered by elements of the IoT infrastructure. Lastly, the researchers intend to analyse if the proposed image clustering approach can be suitably adapted and used in other applications in smart farming, operating auxiliary to decision making in areas such as irrigation/fertilization.

This page was intentionally left blank.

Chapter 3: An easy-to-use application based on evaluating ground soil salinity through the use of UAV images, without using more devices. A use case in Rice farm fields

3.1 Introduction

One of the most significant concerns in the agri domain is the right use of resources, for example: water, fertilizers, soil. Those kinds of resources are direct related to money, for every average farmer. So, they try to use the resources not often but only when needed. If the latter does not happen, the usage of more resources takes place, meaning more spending money, in the other hand, usage of lower than needed water, fertilizers, soil needed may have negative consequences in the yield as someone can understand. In order to take the right decision, there is need for usage of appropriate tools. Concerning that, the current chapter presents an application which was built with the rationale to help the farmers get an assessment of the salinity existing in the soil of their farms via the use of only UAV images, without any use of ground sensors or any other equipment. The main idea was to build a simple interface for non-expert people in computers, so that they could be able to use the image taken from the UAV/drone and with easy handling of the app to get an information for the mean salinity value of their rice farm field.

3.2 Related literature

Extensive research has been taken place in a large number of studies in the agri domain where they use images and photographs so that they can gather, analyze and provision the patterns and how the different kinds of plants behave in relation to their color abridgment and variations. Most of the studies make use of image processing and various statistical schemes on common image's channels, such as Red, Green, Blue (RGB). More specifically, the researchers in [70], used the barycenter of the images in order to compare images coming from different sources via the use of both RGB channels and the RGB color space of chromaticity. The barycenter stands for the center of mass of an object, and in the research, it was used in order to compare the mean intensity of the images come from the two different sources. Via the use of both color space schemes, the researchers could be able to assess the images from various perspectives and proceed to a better understanding of the data.

In another research [70], the color signature of an image can be stood for by three parameters that come from the analysis of how its pixels are distributed on the rgb chromaticity space. The initial parameter is the barycenter, with its parameters: μr, μg, μb, that is computed as the average value of the chromaticity of every pixel and is related to the centre, around which the

other values are dispensed. The next parameter is the variability, with its parameters σr, σg, σb, which is calculated as the SD (standard deviation) of the chromaticity value of all the pixels and stands for the variety of colors in the distribution. Another parameter, the third one, is the Number of Unique Colors (NUC), which is related to the total number of the various locations filled by the distribution in the chromaticity space and stands for the quantity of different colors depicted by the image. Because of the fact that an image's NUC represents an area, it is calculated as a percentage of the whole area of the chromaticity space.

Researchers in [71] study used analysis in images so that they could calculate the three channel colors (RGB) of plant leaves. They produced a .txt file for every image, and each one contained three columns that maintained the following colors: red color (RC), green color (GC) and blue color (BC) of all the pixels in the image. The range of values that they could get ranged from 0 to 255. The authors then computed the average RC, GC, and BC values for every image and used the calculated averages in order to assess their correlation with nitrogen (N) substance in the plant leaves. The data were processed via the use of linear models (such as GLM) procedure targeting to analyzing the variance, the famous ANOVA (ANalysis Of VAriance). The means were compared using the test of Tukey's, something most of the times used in statistical analysis for multiple comparisons.

In another research study [72] the authors made use of sensor which operate in hyperspectral and multispectral range, so that they could evaluate soils that were affected by salinity in rice cultivations. They realized SMA, which stands for Spectral Mixture Analysis to aggregate data from Operational Land Imager - OLI/Landsat-8 so that they could estimate the soils. They contact measurements in salinity via the use of EC sensors, which means Electrical Conductivity, of samples gathered from the soil across 53 different areas, and then classified them to saline and non-saline. They processed data from the Thematic Mapper/Landsat-5 in order to indicate the NDVI (Normalized Difference Vegetation Index) variations in a range between the years 1984 to 2022, at the locations. Through the use of OLI (Operational Land Imager)/Landsat-8 and of course the Hyperion/Earth Observing One, they ended up in 5 indices related to salinity and scores from Principal Component Analysis (PCA) that there realized to the pixels that represented soil values. The indices along with PC1, better known as first principal component, were exploited with the aim to assess soil salinity by using regression.

In research realized by [73], the authors estimated the salinity infiltration in the Tra Vinh Province, an area part of Mekong Delta of Vietnam. They made use of images coming from Landsat 8 OLI so that they could retrieve indices for soil salinity assessment, having in mind the following VIs:

a) VSSI (Vegetation Soil Salinity Index)
b) SAVI (Soil Adjusted Vegetation Index)
c) NDVI (Normalized Difference Vegetation Index)
d) NDSI (Normalized Difference Salinity Index)

They executed a statistical analysis between soil salinity (EC) and the related Vegetation Indices that they exported from Landsat 8 OLI images. They resulted in that the NearInfraRed (NIR) band and VSSI VI presented better correlation than the other indices. New comparisons, output that soil salinity exported from Landsat 8 was precise with determination coefficient, R^2 = 0.89, RMSE = 0.96 dS/m as far as NIR band is concerned, and R^2= 0.77, RMSE = 1.27 dS/m for VSSI Vegetation Index. What researchers found is that Landsat 8 OLI images present high contingent for time and space in magnitude of soil salinity at the higher level of soil stage.

3.3 Introduction to linear regression and multiple linear regression

3.3.1 Linear Regression basics

Linear regression is a technique in order to model and analyze data for predicting (future) values. The simple linear regression, there is the generation of a bivariate model in order to predict a future variable (*y*) given and input (*x*) [74]. The famous linear regression scheme forecasts the connection between an output variable *y* and a single interpretative variable *x*, given that there is a number of data which contains values of both *x* and *y* values for a certain sample. Let's have the following *x* (NDVI*), y* (salinity) values (**Table *8***), and the related graph (**Figure 69**).

salinity	NDVI
0.487	0.706824
0.783	0.764273
0.8165	0.805524
0.691	0.806802
0.8915	0.837463
0.977	0.860352
0.6885	0.817152
0.6825	0.713621
0.328	0.60501
0.297	0.525239
0.243	0.447146
0.434	0.565821
0.386	0.639362

Table 8 Linear Regression X (NDVI), Y (salinity) values.

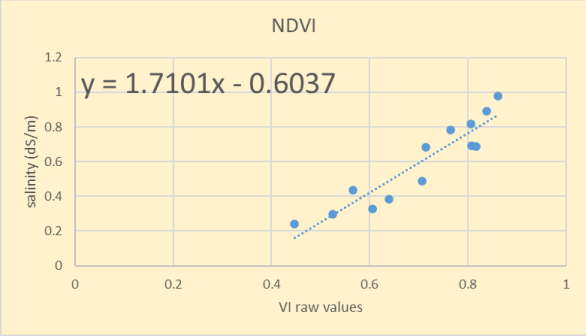


Figure 69 Graphing example of a linear regression equation. On x-axis are the VI (Vegetation Index) values, here are the NDVI, and on y-axis are the salinity values.

As it is obvious in **Figure 69**, the equation $y = 1.7101 \cdot x - 0.6037$ can predict *y* values (salinity) for any *x* value in the range, thus making a prediction for values that are not presented in the **Table 8**. In order to fit a straight line to the various points ("dots") on the above plot, there is use of linear regression, which is the equation of this line. Generally speaking, the equation for the line in regression scheme obeys to the following scheme[35]:

$$y = a + \beta \cdot x$$

[35] https://en.wikipedia.org/wiki/Simple_linear_regression

where:
$\boldsymbol{\alpha}$: is the constant, the point where the line "cuts" the *y* axis of the graph.
$\boldsymbol{\beta}$: stands for the slope of the line, which basically means how much the *y* value rises, when there is a one-unit increase in *x.*

3.3.2 Multiple Regression basics

Regression analysis takes place in order to determine the relation between two or more parameters that contains cause-effect relations and to generate forecasting for the scheme by making use of the relation [75]. The regression which uses only a single independent variable is named simple regression analysis, whereas the analysis that incorporates more than a single independent variable is named multiple regression analysis. Via simple regression analysis, the connections between a dependent variable and an independent variable undergo analysis, and the equation that depicts the linear connections between independent/dependent parameters obey to one or more equations. It is widely known that regression models consisted of one dependent variable and many independent variables is named multivariable regression analysis. Multivariate regression analysis is given by the following equation:

$$y = \beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_n \cdot x_n + \varepsilon$$

where:
$\boldsymbol{y}$ stands for dependent variable
$\boldsymbol{X_i}$ stands for the independent variable
$\boldsymbol{\beta_i}$ stands for the parameter
$\boldsymbol{\varepsilon}$ represents the error

3.4 Evaluation and analysis of the proposed application

The current chapter presents an application that was built in order to assess the salinity of UAV .tif image that displays farm field, in a very plain way, able to be managed by a user that does not have special knowledge in using a computer.

The application was made using python script and the following libraries/packages:
a. ConfigParser
b. GDAL
c. matplotlib
d. numpy
e. opencv_python
f. Osgeo
g. Pandas
h. PyQt5
i. Rasterio
j. sklearn
k. Tifffile

It uses image processing so that it can calculate the various Vegetation Indices by processing each image's bands and assessing the salinity via the use of special mathematical models. The way it works internally will be presented in later paragraph in the current chapter.

So, the procedure is the following: the user chooses from the "Calculate" menu which VI fits its situation according the experience he has, because each VI uses different bands. In **Figure 70**, a BGR UAV image is used, where the letters come out from the words: B = Blue color, G = Green color, R = Red color. The "Calculate" menu contains 9 operations, each one assigned to the calculation of a Vegetation Index. In the parentheses, someone can see what type of image is needed, so that the application can calculate the corresponding VI. This is the rationale behind the existence of 2 buttons on the right-side top side of the application. The left button is used in order for the user to choose a BGR UAV image, whereas the right button is used to load a GRRN UAV image, where the letters GRRN stand for Green, Red, Red-Edge, NearInfraRed. So, according to which types of bands the image contains, the users use the respected button (Left one or Right one).

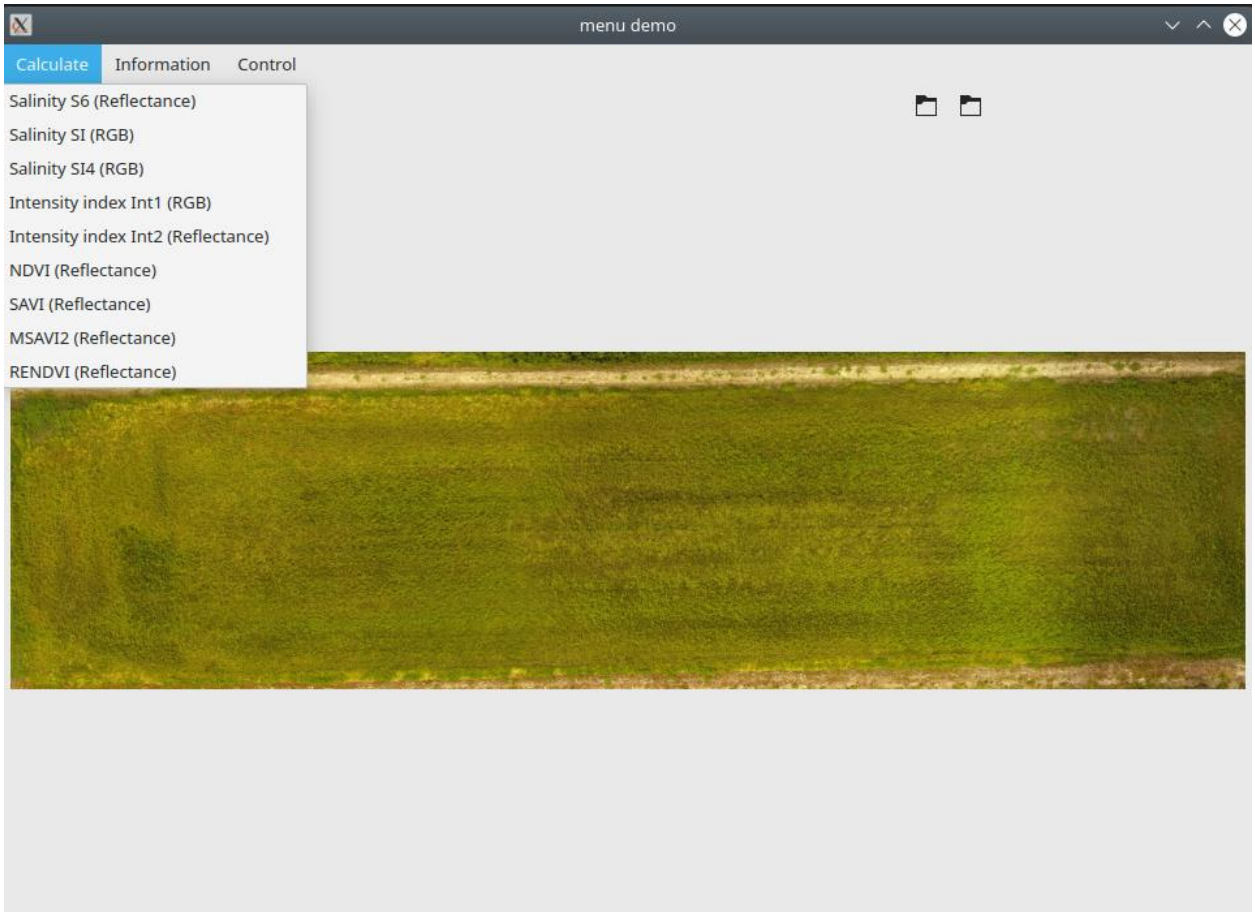


Figure 70 Application for evaluating soil salinity, where the user needs to choose either RGB or Reflectance UAV images.

The application is configured with a Linear Regression model and can assess a range of salinity, measured in deciSiemens/meter (dS/m) of the chosen image, which depicts the rice field. Thus, the farmer can decide if there is need to irrigate their farm field in order to mitigate water

spending and secure their yield from irreversible damage. Otherwise, when the water (soil) salinity levels in the rice field belongs to the desired range the farmers can postpone an irrigation, save in that way resources and money. As someone can observe in **Figure 71**, the application outputs an application of the salinity in the fed UAV image.

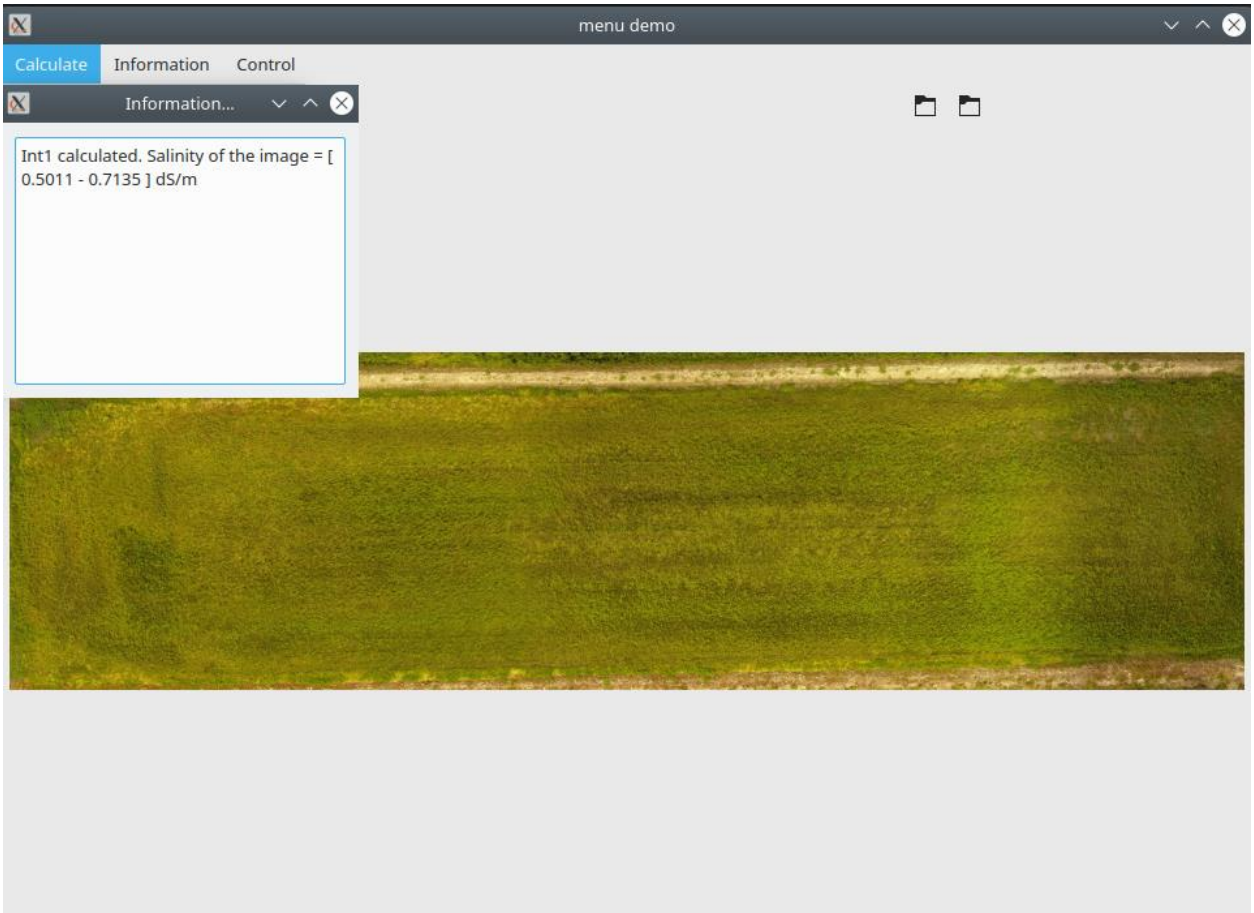


Figure 71 Assessed salinity range for the selected BGR UAV image. As it is obvious the Int1 Vegetation Index was used.

3.5 Data gathering and analysis

The current sub-chapter analyzes the levels of salinity in a rice field using data aggregated from two ground sensors positioned in different areas of the rice field in years 2021 and 2022. So, to obtain data quality many parameters were implemented to the dataset. First of all, any rows with absent data for the salinity variable were not used in the study. Secondly, negative or zero values related to the salinity variable were considered non-logical and so not used in the analysis. Also, salinity values less than 0.01 were regarded as unreliable and not used in the analysis.

After the stage of applying the previous parameters, there were collected 55,881 points of data for both years, that contained measurements of the level of salinity, water level, water temperature and water content. More especially, for the year 2021, 31,614 data points were gathered from the two nodes during June, July, August and September. In the year 2022, 24,2767 data points were gathered from the two nodes for the months June, July, August, September, October, without data related to water level from node 3 in 2022 because it did not operate due to sensor error. **Table 9** depicts the summary of the statistics of all the parameters' values for the year 2021, whereas the **Table 10** displays the same parameters' values for year 2022. Both tables are related to the two ground sensors.

	SALINITY		WATER LEVEL		TEMPERATURE		CONTENT	
	2021	**2022**	**2021**	**2022**	**2021**	**2022**	**2021**	**2022**
MEAN	7.7341	6.0819	26.313	20.9485	20.9275	21.5689	2794.57	2825.55
VARIANCE								
STAND DEV	4.039	1.627	7.65	10.39	5.20	3.89	442.62	133.16
MIN	1.93	3.09	1.9	0.00	8.10	10.30	1867.24	2416.91
25%-PERC	4.13	4.25	24.62	10.53	16.80	19.10	2566.40	2709.76
50%-PERC	7.63	6.46	28.6	23.60	21.40	22.30	2993.62	2857.34
75%-PERC	9.31	7.28	31.7	29.84	24.60	24.40	3136.76	2945.92
MAX	18.85	9.87	52.11	44.92	46.50	36.60	3269.16	2990.04

Table 9 Overall statistics for both ground sensors' data in year 2021.

In order to have better understanding of the aggregated data, there was a comparison between the locations of where the two sensors were placed. Extended analysis showed that node 3 that contained higher levels of salinity in comparison to node 1 in years: 2021 and 2022, as it is depicted by the means values in **Table 10**. Moreover, the levels of ground salinity were higher in 2022 than in 2021.

The aforementioned results are also supported by T-tests, that took place for every occasion, with p-values less than 0.05, and which indicate rejection of the null hypothesis. As a conclusion, there can be said that there is a serious difference in the salinity values between the two sensor locations, where node 3 indicates often higher salinity values than node 1. This information is essential concerning the construction of effective irrigation system in order to keep best salinity levels in the rice farm field.

Variable	**Means**	**Standard Dev**	**T-test statistic**	**p-value**
Salinity 2021_overall	7.7341	4.0390	66.113	0.00
Salinity 2022_overall	6.0814	1.6256		
Salinity 2021_node1	6.1489	2.4632	-74.726	0.00
Salinity 2021_node3	9.2434	4.6259		
Salinity 2022_node1	5.5376	1.2238	79.58	0.00
Salinity 2022_node3	6.8145	1.6398		

Table 10 T-test related to salinity for years 2021, 2022 for both the two ground sensors.

The observations are supported with auxiliary histograms, depicted in **Figure 72** and **Figure 73**, where someone can view the differences in salinity levels between the two ground sensors in years 2021 and 2022.

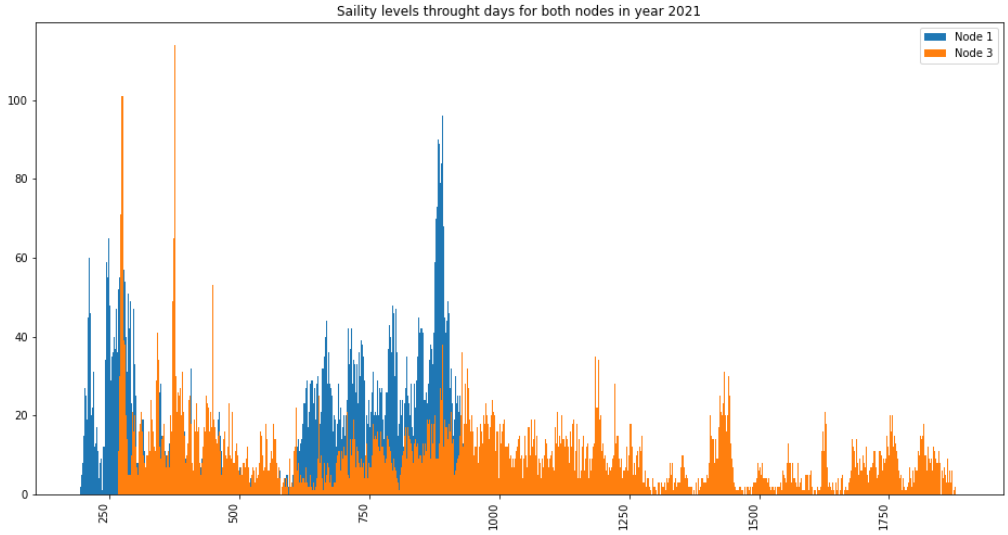


Figure 72 Salinity histogram for year 2021.

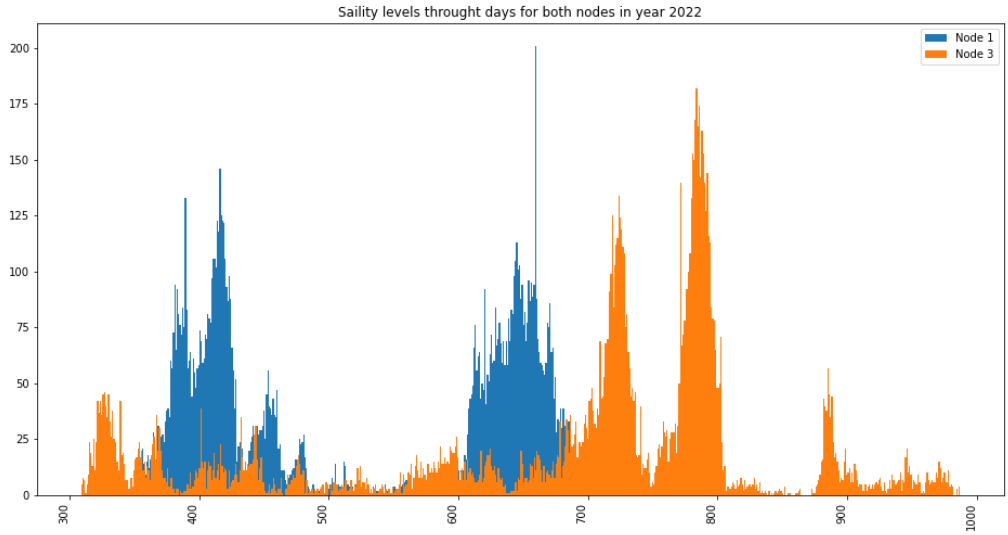


Figure 73 Salinity histogram for year 2022.

Moreover, further analysis indicates a stronger correlational between salinity and water content values. Spearman's correlation coefficient places this value to 0.761 for the latter pair in both years 2021 and 2022. On the other side, there was no serious correlation identified between the salinity with either water temperature or water level. The result of these findings show that water content is the key in defining levels of salinity in the rice field and can help as an essential variable for more analysis.

	Water Level	Water Temperature	Water Content	Salinity
Water Level	1.00	0.083	-0.084	-0.083
Water Temperature	0.083	1.00	-0.352	0.367
Water Content	-0.084	-0.352	1.00	0.761
Salinity	-0.083	0.367	0.761	1.00

Autocorrelation and stationarity should be analyzed, since the data for the salinity was gathered over a long period of time. The analysis showed that Durbin-Watson statistic test concerning salinity information aggregated from both sensors in both years, 2021 and 2022, output values close to zero. This event, defines high positive correlation. The implemented Dickey-Fuller test, that took place in order to analyse the stationarity of the salinity data, did not succeed in rejecting the null hypothesis of non-stationarity, so, someone can understand that it lacks stationarity.

The existence of high positive autocorrelation and non-stationarity in the data related to salinity could have consequences to the validity of the current analysis and the respected conclusions coming from the data. At this point, it is essential to have in mind these factors when illustrating the results and going deeper to research hypotheses. More analysis using time-series modelling schemes can be needed in order to effectively figure the possible effects of these factors related to data for salinity. Taking into consideration the aforementioned, the output is the following regression scheme, given by the formula:

$$salinity = -7.957 - 0.016 \cdot water\ level - 0.074 \cdot temperature + 0.006 \cdot water\ content$$

Based on the above information, it seems that the regression model maybe not a good match for the data. The R-squared value of 0.378, which is considered as low, shows that only 37.8% of the variability in the salinity, which is the dependent variable, is explained by the following: water level, temperature, water content, which are considered as independent variables. This implies that maybe there are parameters that are affecting salinity, however they are not captured by the model.

Moreover, the low value of Durbin-Watson statistic related to residuals, 0.007, shows the existence of autocorrelation, that violates the initial case of residuals that were independent. The latter indicates that the proposed model does not capture all the information that exist in the data. Also, the p-value of F-statistic of 0.00 reveals that at least one independent variable is seriously connected to the dependent variable, but only this cannot result in a good match. The low p-value connected with the JB (Jarque-Bera) test of 0.00 reveals that there is not normally distribution of the residuals, something that violates the initial hypothesis of linear regression, where the residuals are distributed normally.

3.6 Image Analysis

Apart from the sensor data analysis, the current chapter delegates also gathered aerial photographs captured by a drone. The photographs gave information concerning RGB (Red-Green-Blue) and GRRN (Green-Red-RedEdge-NearInfraRed) levels of channels or bands as they are called. The images were stored in .tif file format and were represented in 2x2 matrix for every

channel. Every cell depicts a pixel of the image. A total of 27 photographs were stored on different days, between the years 2020 and 2021.

In research [76] the authors interpret than when image analysis takes place, the pixels in the image are developed into a data matrix. Every row stands for a pixel, whereas the columns stand for the color or the spectral bands. The latter lets for building multivariate projection models such as PCA, which stands for Principal Component Analysis, or PLS, which stands for Partial Least Squares. The selection between PCA or PLS will be based on the nature of data and the targets of the analysis. Via this approximation, it is possible to implement multivariate image analysis, that can output awareness into the connections between different spectral or color bands and the various image's characteristics are analyzed.

In order to support the current study, the BGR and GRRN channels, captured in the images, were used with the aim to compute nine Vegetation Indices. Those levels were:

$$SI = \frac{blue + red}{2}$$

$$SI4 = blue * \frac{red}{2}$$

$$Int1 = \frac{green + red}{2}$$

$$int2 = \frac{green + red + nir}{2}$$

$$RENDVI = \frac{nir - red\ edge}{nir + red\ edge}$$

$$S6 = red * \frac{nir}{green}$$

$$NDVI = \frac{nir - red}{nir + red}$$

$$SAVI = \frac{3}{2} * \frac{nir - red}{nir + red + 0.5}$$

$$MSAVI2 = \frac{2 * nir + 1 - \sqrt{(2 * nir + 1)^2 - 8 * (nir - red)}}{2}$$

For every channel, the averages of all images where computed, resulting in 27 sets of averages, one fitted for each image. Moreover, the SDs were calculated for every level. Pearson's correlation showed that every band/channel/level had an increased correlation between them, apart from RedEdge, NIR and Int1 Vegetation Index.

3.7 Single Regression Analysis

Based on the data gathered both from UAV .tif images and sensing data from the sensor modules, the procedure is as follows: At first, the Vegetation Index on every image was computed in relation to the equations presented previously. Focus was paid in the following Vegetation Indices: S6, SI, SI4, Int1, Int2, NDVI, SAVI (Soil-Adjusted Vegetation Index), MSAVI2 (Modified Soil Adjusted Vegetation Index - 2), RENDVI (Red Edge Normalized Difference Vegetation Index). So, for every UAV image, there was a calculation of 9 Vegetation Indices. Moreover, for every UAV image a salinity number was fit. The salinity number was the result of the 2-salinity sensor positioned on ground: node_1 and node_3, and especially the average value of the 2 salinity sensor nodes on the timestamp that the image was taken from the drone. For every VI there was a fit between the VI values and the corresponding salinity values. For example, for S6, on x-axis there were specified the various VI values and on y-axis there were specified the corresponding salinity values. Via the use of linear regression an estimating line was picked. That line/curve can be used for forecasting values. For instance, given a new VI the curve can map this VI value to a salinity value. **Table 11** presents the various mapping between salinity values and Vegetation Indices.

datetime	salinity	S6	SI	SI4	Int1	Int2	NDVI	RVI	SAVI	MSAVI2	RENDVI
07-12-21	0.487	0.22	0.054	0.002	0.069	0.236	0.707	6.755	0.455	0.455	0.088
7/17/2021	0.783	0.198	0.039	0.001	0.052	0.207	0.764	8.278	0.464	0.464	0.099
7/22/2021	0.817	0.216	0.039	0.001	0.049	0.223	0.806	10.16	0.516	0.532	0.156
7/27/2021	0.691	0.238	0.036	0.001	0.051	0.238	0.807	9.956	0.536	0.558	0.128
08-01-21	0.892	0.228	0.035	0.001	0.048	0.245	0.837	12.195	0.568	0.603	0.16
8/16/2021	0.977	0.216	0.032	0.001	0.048	0.26	0.86	14.203	0.602	0.651	0.141
8/31/2021	0.689	0.241	0.041	0.001	0.06	0.275	0.817	10.555	0.582	0.617	0.127
09-10-21	0.683	0.282	0.056	0.002	0.078	0.274	0.714	6.56	0.499	0.507	0.101
9/15/2021	0.328	0.326	0.072	0.003	0.096	0.279	0.605	4.425	0.423	0.416	0.092
9/20/2021	0.297	0.321	0.08	0.003	0.104	0.267	0.525	3.549	0.355	0.34	0.08
9/25/2021	0.243	0.394	0.099	0.005	0.125	0.297	0.447	2.893	0.317	0.303	0.091
07-07-22	0.434	0.26	0.074	0.004	0.088	0.24	0.566	5.503	0.361	0.357	0.106
07-12-22	0.386	0.229	0.054	0.002	0.071	0.222	0.639	6.866	0.399	0.399	0.093

Table 11 Mapping between Vegetation Indices and the related salinity values.

For every VI a special equation is mapped. All of them are seen below. The output for every Vegetation Index is the following:

1. (S6): y = -3.1588x + 1.411
2. (SI): y = -10.485x + 1.167
3. (SI4): y = -156.56x + 0.8839
4. (Int1): y = -8.6789x + 1.2199
5. (Int2): y = -3.3747x + 1.4396
6. (NDVI): y = 1.7101x - 0.6037
7. (SAVI): y = 2.3571x - 0.5093
8. (MSAVI2): y = 1.9539x - 0.3395
9. (RENDVI): y = 7.384x - 0.2384

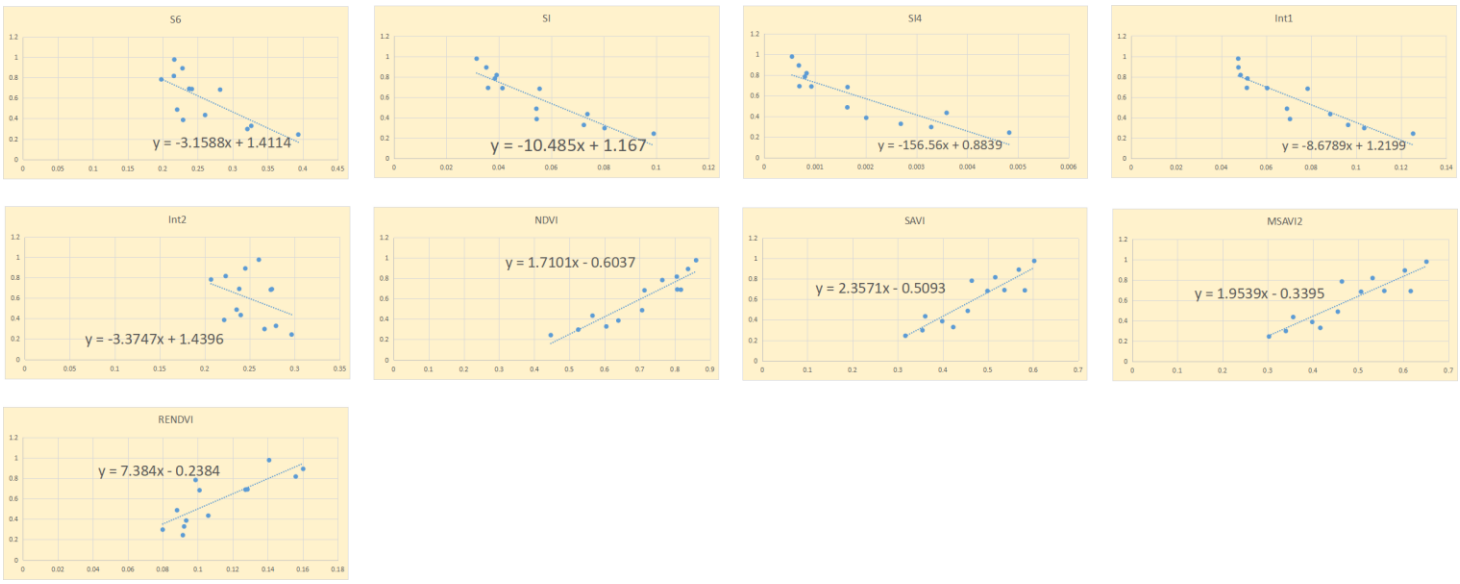


Figure 74 Linear Regression mapping to the related salinity values, for all 9 Vegetation Indices.

3.8 Further general Analysis

The dataset is consisted of a matrix with 27 rows x 14 columns, where every row displays the image captured by a drone on a specific timestamp, and every column demonstrates a different level computed based on BGR, GRRN and the remainder chromatic levels of the image. The averages of all pixels for every photograph and level were computed, and the average salinity of the specific timestamp the photograph was captured was subjoined to evaluate the effect of the levels of salinity on the rice plant observed in the images. The data were processed via the use of multilinear regression by using Python script, ending up in a regression line that displays the connection between the averages of every day and the result of salinity of every rice plant.

The regression scheme was processed with salinity, water level, water temperature and water content, representing the dependent variables and averages from every level of the images, representing the independent variables. Nevertheless, because of the fact that data were narrow, the averages of salinity from one day before and two days before the timestamps the images were captured, were added in the model to increase the strength of the model. The outcome was that four linear schemes were matched and their results can be observed below in the following tables.

The R-squared value equals 0.83 is translated to the fact that 83% of variation in the dependent variables can be analyzed by the variables that are independent in the regression model. The F statistic p-value of 1.24-e8 indicates that the regression model is statistically notable with a great level of confidence, showing that at least one of the independent parameters (variables) has serious consequences on the dependent variable. The weak covariance type shows that the matrix consisted of covariances was assessed using the straightforward maximum likelihood method, supposing normally distributed errors that contain stable variance. The DW (Durbin Watson) test statistic cost guides to the fact that no autocorrelation exists in the residuals, and the p-value linked with the JB test statistic points out that the residuals are normally distributed, and there is no way of rejecting the null hypothesis of normality consisted of 0.05 level of

attention. Nevertheless, the rest of 17% of variability may be a result of other considerations not contained in the model related to measurement errors.

	Coefficient	Standard Error	t	p-value (sig.)
Constant	814.3753	449.121	1.813	0.08
Red	-2389.4529	902.45	-2.648	0.013
Green	2125.2142	570.118	3.728	0.001
Blue	-628.3513	366.29	-1.715	0.097
Red_edge	-907.2423	503.949	-1.8	0.082
NIR	1025.4111	540.399	1.898	0.067
SI	-1508.9021	438.01	-3.445	0.002
SI4	2.45E+04	9490.53	2.583	0.015
Int1	-132.1194	304.978	-0.433	0.668
RENDVI	-245.139	262.995	-0.932	0.359
S6	135.1502	107.402	1.258	0.218
NDVI	-474.8017	325.717	-1.458	0.155
SAVI	2834.191	1737.968	1.631	0.113
MSAVI2	-1332.6408	805.254	-1.655	0.108

Table 12 All the data related to regression of salinity with the different Vegetation Indices.

In **Figure 75** someone can observe a comparison between the actual and predicted salinity values as a result of the regression method we followed.

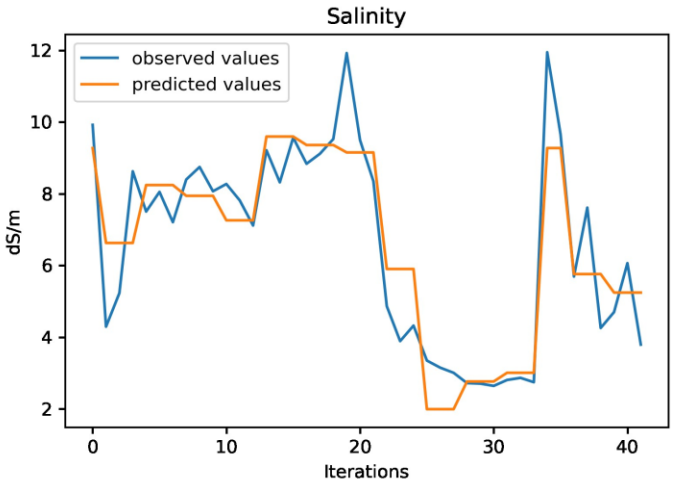


Figure 75 Comparison of predicted and observed salinity values extracted from **Table 12** above.

The regression model of the next case gives an R-squared of 0.834, as it is presented in **Table 13**. This shows that part of the variation (in the dependent variable) can be explicated (by the independent variable/s). The F-statistic outputs a p-value of 7.11e-07, something which shows that the total model supports a better match to the data than a mitigated model. The Durbin-Watson test statistic gives value equal to 2.479, which does not support the existence of the autocorrelation in the remaining of the scheme, albeit it may not track autocorrelation in higher-degree. As far as the JB test is concerned, it outputs as p-value the number 4.50e-19, showing strong results against the null hypothesis related to normality and so the residuals of the scheme are not distributed normally. The model supports a good matching with the data and it is statistically important, however, it is considerable to underline the contingent for non-normality in the residuals.

	Coefficient	Standard Error	t	p-value (sig.)
Constant	-6045.4236	1973.695	-3.063	0.005
Red	-1.97E+04	4610.605	-4.263	0
Green	1.63E+04	4127.9	3.954	0.001
Blue	-1.07E+04	3237.422	-3.306	0.003
Red_edge	1786.0172	803.49	2.223	0.036
NIR	-6571.8054	2572.49	-2.555	0.017
SI	-1.52E+04	3655.528	-4.152	0
SI4	5.95E+05	1.51E+05	3.932	0.001
Int1	-1666.1406	924.735	-1.802	0.084
RENDVI	3040.3362	776.422	3.916	0.001
S6	2420.803	537.399	4.505	0
NDVI	4102.248	1209.664	3.391	0.002
SAVI	-3.60E+04	9367.067	-3.848	0.001
MSAVI2	1.53E+04	4363.427	3.511	0.002

Table 13 All the data related to regression of water level with the various channels of the image.

In **Figure 76** someone can observe a comparison between the actual and predicted water level values as a result of the regression method we followed.

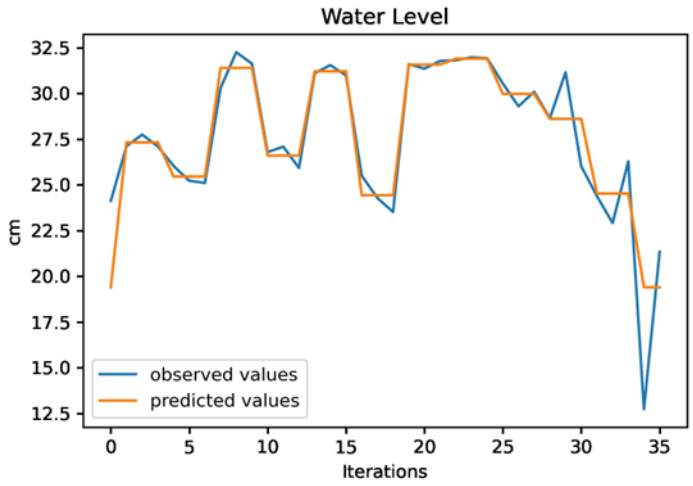


Figure 76 Comparison of predicted and observed water level values extracted from **Table 13** above.

The R-squared value of 0.666 (**Table 14**) shows that the independent variables appearing in the regression model, interpret 66% of the variation in the dependent variable. F-statistic p-value equals 0.543, something which indicates that there is no considerable linear linkage between independent and dependent variables. The DW statistic equals 0.937, and shows a probable positive autocorrelation in the remaining of the model. Moreover, the p-value of JB statistic is very small, something which shows that the residuals are distributed following a normal distribution. Thus, a further analysis and more tests are needed in order to have a clear view of the regression model.

The R-squared value of 0.929 (**Table 15**) states that the independent variables in the regression scheme interpret the 93% of the variation displayed in the dependent variable, letting 6% of the variation not interpreted. The F-statistic's p-value = 0.00, that is under the straightforward used level of 0.05 significance. The latter shows that there is an intense relationship between dependent and independent parameters. DW gives value of 2.05, a number close to 2, something that reveals no significant autocorrelation in the remainder of the regression model. The p-value is connected with the JB statistical test, which has the value of 0.77, more than the ordinal used significance level of 0.05, revealing that the remainders probably follow normal distribution, so the null hypothesis cannot be rejected on 0.05 level of significance. However, the results support a very good starting point for more analysis and propose that remote sensing is a very useful means for monitoring and controlling soil salinity in rice fields.

	Coefficient	Standard Error	t	p-value (sig.)
Constant	-749.7022	945.414	-0.793	0.434
Red	-75.3785	1899.685	-0.04	0.969
Green	-2089.6738	1200.117	-1.741	0.092
Blue	1362.6904	771.052	1.767	0.087
Red_edge	392.9623	1060.829	0.37	0.714
NIR	-75.3492	1137.557	-0.066	0.948
SI	643.6559	922.025	0.698	0.49
SI4	1.54E+04	2.00E+04	0.769	0.448
Int1	-1082.5261	641.988	-1.686	0.102
RENDVI	-298.386	553.612	-0.539	0.594
S6	342.197	226.084	1.514	0.141
NDVI	618.9237	685.645	0.903	0.374
SAVI	-4326.9506	3658.478	-1.183	0.246
MSAVI2	1694.9415	1695.086	1	0.325

Table 14 All the data related to regression of water temperature with the various channels of the image.

In **Figure 77** someone can observe a comparison between the actual and predicted water temperature values as a result of the regression method we followed.

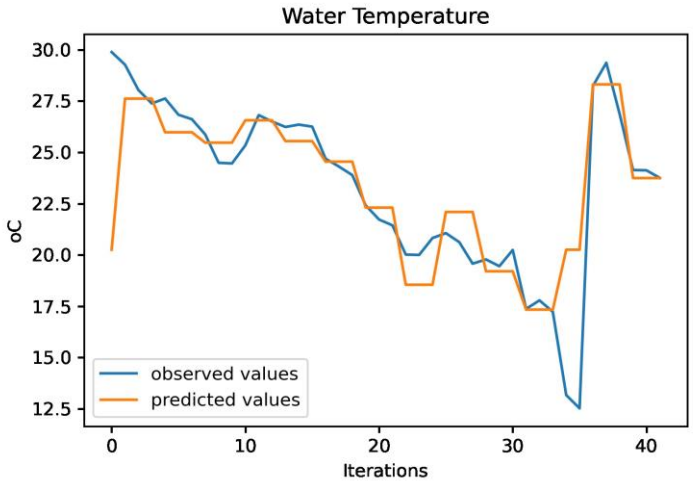


Figure 77 Comparison of predicted and observed water temperature values extracted from **Table 14** above.

The analysis that took place at the beginning, guides to the fact that there is a connection between chromatic levels in images and salinity measured in rice fields. However, extended analysis is necessary in order to bear out this connection and to comprehend better the complex interaction of factors that react on salinity in rice fields. The autocorrelation and the non-stationarity that were identified in the data reveal that more advanced models and mechanisms and probably more data are necessary to track the patterns and their connection with the data.

	Coefficient	Standard Error	t	p-value (sig.)
Constant	6.52E+04	3.76E+04	1.732	0.094
Red	-2.06E+05	7.56E+04	-2.72	0.011
Green	9438.0037	4.78E+04	0.197	0.845
Blue	6.20E+04	3.07E+04	2.019	0.053
Red_edge	7219.0312	4.22E+04	0.171	0.865
NIR	1.12E+05	4.53E+04	2.469	0.019
SI	-7.19E+04	3.67E+04	-1.958	0.06
SI4	1.09E+06	7.96E+05	1.374	10.18
Int1	-9.82E+04	2.56E+04	-3.84	0.001
RENDVI	-8432.75	2.20E+04	0.383	-0.705
S6	2.28E+04	9003.172	2.537	0.017
NDVI	-2.37E+04	2.73E+04	-0.866	0.393
SAVI	1.25E+05	1.46E+05	0.858	0.398
MSAVI2	-9.45E+04	6.75E+04	-1.4	0.172

Table 15 All the data related to regression of water content with the various channels of the image.

In **Figure 78** someone can observe a comparison between the actual and predicted water content values as a result of the regression method we followed.

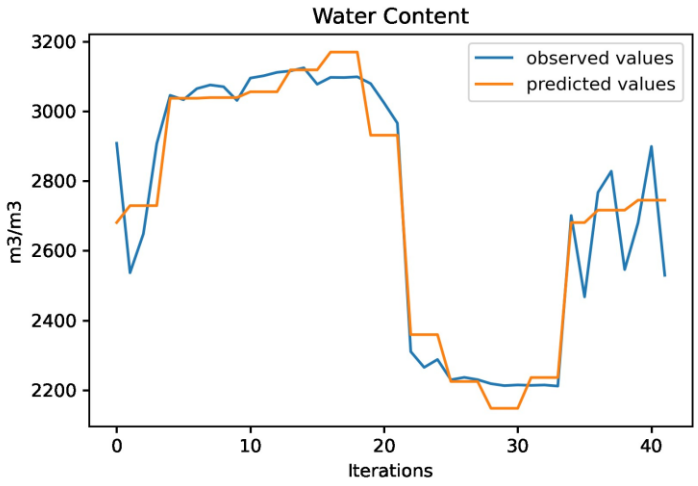


Figure 78 Comparison of predicted and observed water content values extracted from **Table 15** above.

3.9 Conclusions

In the current chapter there was presented an analysis centred on single regression and multiple regression with data collected form sensors put on the ground. Those sensors measure soil salinity, water level, water temperature and water content. The data were analysed from a statistical point of view and some outcomes enabled by the analysis. Single and multiple regression were used with the aim to estimate future salinity values using a built mathematical model. The chapter also delegates single regression through which a farmer with basic knowledge of how to use a computer can feed the model with UAV images either RGB or Reflectance one and assess soil salinity in their rice farm without the need of using auxiliary equipment, for instance ground soil sensors.

For the future, there could be use of a more accurate time-series model for forecasting such as ARIMA (Autoregressive Integrated Moving Average) or SARIMA (Seasonal Auto-Regressive Integrated Moving Average). Moreover, more research should be established in relation to the chromatic levels which are most relative in order to predict soil salinity in rice fields. By solving these issues and enabling more analyses, it may be potent to build a more accurate and effective predictive mechanism in order to monitor and control soil salinity in rice farms.

This page was intentionally left blank.

Chapter 4: Machine Learning and IoT in the agri-domain

4.1 Introduction

Soil Salinity is the process of blending the soil with solvable salts that end in leaving the soil saline. The latter is a major problem in the agri-domain, because salinity mitigates the value of land which affects the productivity. On the other side, saline soils could be the outcome of irrigation, because of the saline containing of the water, and could be from medium to high [77]. Saline soils could be the output of augmented water usage in near coastal fields, due to the penetration of the sea and the flooding that takes place near those areas as an effect of Mediterranean areas' storms. As it is known, in the river deltas, inside the Euro-Mediterranean area, the main crop cultivated is Rice. As a result of the fact that salinity is endemic in fields near cost, rice plants have to be filled with water in order to mitigate water salinity. Thus, very large amount of water is needed to decrease salinity and a significant amount of energy for pumping water from the rivers.

4.2 Related State of Art literature

In [78] the researchers use three Machine Learning (ML) approaches such as the Logistic Regression (LR), Support Vector Machine (SVM) and Random Forest (RF) were tested and contrasted in the identification of salt-affected soils (SAS), in the Raibareli area of India via the use of Landsat 8 OLI/TIRS channels and auxiliary data next to canals and streams which were exploited in order to recognize SAS. Those 3 models were implemented to object-oriented areas produced by those rasters. A total number of 361 areas were used for training, and testing for the RF algorithm. From them the 130 segments were part of the SAS and the rest 231 depicted other features and normal soils. In their approach, the researchers used the 70% of the data for the training session and the rest 30% for the testing session. They achieved 96% accuracy concerning LR model, 98% accuracy for the SVM algorithm and 98% accuracy for the RF model. All the latter concerning the testing session. The outcomes were verified with in-field observations, and high-resolution data coming from Google Earth database. The end result showed that the aforementioned 3 different models could recognize 31,954, 16,679 and 14,070 ha areas respectively in Raibareli distinct in India.

In [79] the researchers mix Sentinel-2 Multispectral Imager (MSI) data and MSI-derived covariates with sensed soil salinity and to implement 3 different ML algorithms in order to assess and map the soil salinity in the targeted area. In relation to the known transportation conditions, the interesting area and the quadrat were fixed, and the five-point method was used in order to gather the soil mixed samples, where around 160 mixed soils from the soil were picked. The famous Kennard-Stone (K-S) model was exploited in order to classify the data. The 70% of the data were used for training and the 30% of the data were used for testing. The following ML models were used: Support Vector Machines (SVMs), Artificial Neural Network (ANN) and Random Forest (RF). Among their many outcomes, one very interesting is the following: the mean reflectance of each band of the MSI data is in the range of 0.21 to 0.28. In response to the spectral

characteristics that behave to various soil electrical conductivity (EC) levels in the range 1.07 dS/m to 79.6 dS/m, the reflectance of the soil containing salinity is in the range 0.09 to 0.35.

In [80] the authors used Unmanned Aerial Vehicles (UAVs) for remote sensing in order to measure salt quantity in quinoa plants. Three different UAV sensors were exploited: 1) a WIRIS thermal camera, 2) a Rikola hyperspectral camera and 3) a Riegl VUX-SYS Light Detection and Ranging (LiDAR) scanner. They evaluated many vegetation indices, canopy temperature and plant height via remote sensing. Moreover, they sensed their relation with ground assessed parameters such as salt treatment, stomatal conductance and the real height of the plants. The outcomes indicate that widely used multispectral Vis (Vegetation Indices) are not the correct tool in order to discriminate between salt affected and manage quinoa plants. As they claim, the Physiological Reflectance Index (RPI) executed best and showed an obvious separation between salt affected and control plants. The use of LiDAR also showed an obvious distinction, since salt treated plants were on average 10 cm shorter than control ones. Another parameter seriously induced was the canopy temperature. The latter needed one more step in achieving that result: an NDVI (Normalized Difference Vegetation Index) clustering. This step guaranteed comparison of temperature for similar vegetated pixels. Data mixing of all three sensors in MLR (Multiple Linear Regression) model increased the prediction power and for the total dataset R^2 =0.46, with a few subgroups approaching R^2 = 0.64. The authors claimed that remote sensing via UAV is very useful for identifying and evaluating stress in a yield, moreover by using multiple measurement approaches can help in order to increase the accuracy.

4.3 Rice water quality enabler

One way in order to continuously sampling the salinity, is through IoT sensors, positioned in the ground and renew the water in the farm field when it is only necessary. However, when the rice field is huge, as it occurs in real situations, it is very expensive for the end user or the farmer to place many IoT sensors to their farm field. Moreover, it can be a problem when agriculture machinery has to be placed in the farm fields, such as tractors. For this reason, it is urgent need to monitor salt stress in the field without sensors placed on ground, but indirectly, via processing of UAV and satellite image, which is an indirect method. So, as referred before, one (expensive) solution to measure soil salinity could be to place many IoT sensors in many places inside the rice farm field. A (non-expensive) solution would be the following: if the plants in a rice pad are covered with increased value of salinity that is measured by an IoT sensor, then all the near rice pads would face the same soil salinity stress. This can be analyzed by UAV or satellite images, remotely, without need to engage IoT sensors.

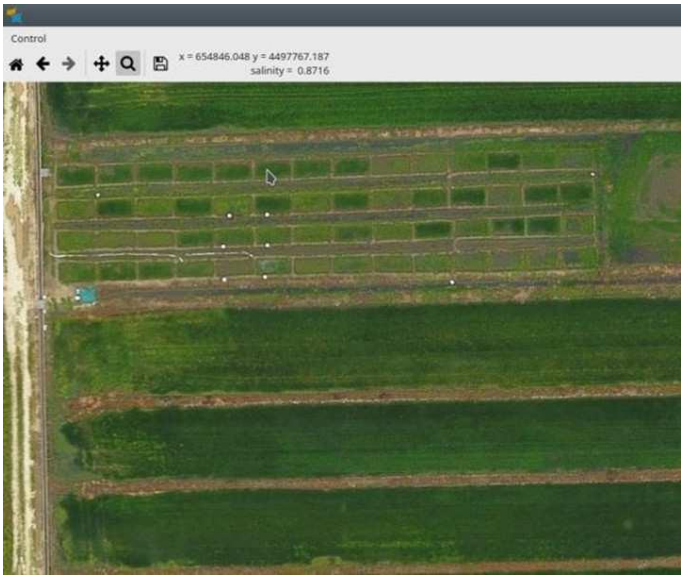


Figure 79 Measured electrical conductivity and related coordinates while pointing the area with the mouse pointer[36].

Figure 79 depicts an example of an Optimal Water Quality Enabler. The users input a UAV image captured from a drone. Then they choose the time range that has to be relevant with image captured, and the algorithms matches the timestamp of the image capture with the measurements from the ground sensors. The user picks one of the 2 IoT ground sensors in order to be used as main reference for the image connection. What enabler does is to output a value on the application with the coordinates LOGITUDE, LATITUDE with the related soil salinity estimation, according to where it places the mouse pointer. The application uses Inverse Distance Weighting method and the related IoT sensor node placed in the soil.

By using this kind of soil salinity estimation, the farmer or the end user is able to take decisions about when to place water to their rice farm field, even in cases where the field is not equipped with IoT sensor. The rice farm field depicted in **Figure 79** contains only two IoT sensors for measuring the soil salinity. Also, the farmer avoids damages in his crop yield because it gets informed very quickly about soil salinity increases, and they save precious water that in those conditions and for the amount they need it, costs a lot.

4.4 Maize irrigation Enabler

The optimal management targeting irrigation of maize farm fields is related to weather related to weather conditions. That is why the Maize Irrigation Enabler supports weather predictions in local level, making good use of the most modern regression techniques. To analyze it further, the described enabler of the current section utilizes RNN-LSTM (Recurrent Neural Network – Long Short-Term Memory). The RNN consists of 2 inputs. The first input is linked to the present and the second input is linked to the past. The output of every step is positioned as input to the following step and looks like a feedback loop. The RNN includes an inner state which is accountable for computing the data sequence existing in input, that is used for griping the new data flowing a recursive logic. The biggest drawback of RNN scheme is the fact that they cannot face the gradient

[36] https://github.com/Axel-Erfurt/OrthoViewLite

problem, which has the consequence that it's not able to sense the reliance over large time history. That happens, as a result of the fact that RNN scheme while undergo training is controlled by the most recent data [81]. A solution to this problem was given by adding the LSTM core in RNN. LSTM makes use of a memory mechanism, whereas a RNN uses neurons. LSTM supports quick training and it is in position to learn good enough through the use of what is called as a continuous short-term memory, being able to manage and store large results of time-series steps [82].

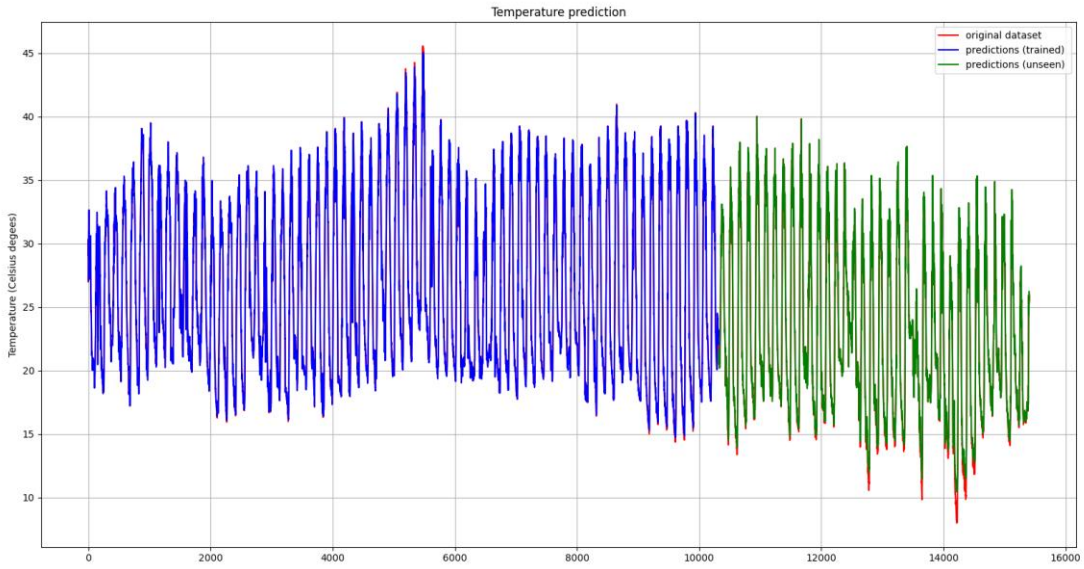


Figure 80 Temperature forecast via the use of an RN-LSTM model.

The implementation of the current enabler is an application developed in python programming language, which collects data from a meteorological station placed near the target maize field. An RNN-LSTM model has been developed in order to predict key weather parameters such as wetness, temperature, and RH (Relative Humidity) gathered from past weather data, where every model links to each parameter. An idea of temperature prediction is depicted in **Figure 80**. Those kinds of forecasts are then exploited for providing the most suitable irrigation handling by feeding with the appropriate input a DSS (Decision Support System) enabler. The maize enabler described in the current section supports weather predictions that apart from maize irrigation, could also be used in other applications.

4.5 Optimal water quality via the use of Convolutional Neural Network

The aim of this section is to present how a CNN Machine Learning model can aim the end user to estimate the soil salinity in a rice farm field, which is very significant information for a farmer. The farmers must have an estimation of the rice farm for 2 reasons:

1. The first reason is that they have to keep the soil salinity inside a specific range, otherwise if it excels a specific level, it may destroy their crops
2. The second reason, is that the precise irrigation in their rice farm field and not the unneeded irrigation saves money, because water in such large volumes is costly for a farmer.

The whole implementation was based on satellite images depicting rice farm fields and measurements gathered from IoT sensors placed inside the farm plot. The IoT sensors gathered information in a continuous frequency, sensing soil salinity.

Initially there was a linkage between the datetime that the satellite images were captured and the ground IoT sensors measurements. We kept the values of the IoT soil salinity sensors placed in the rice farm field. As it is shown in **Table 16** every day of the captured satellite image was connected with the mean salinity sensed from the IoT devices. Consequently, each salinity value was rounded to a value (as presented in column 3 of the **Table 16**). Every new (rounded) value maintains a range of ± 0.05. For instance, the photo captured on 2021-07-12, has a salinity of 0.487, and after rounding its new value is 0.5 ± 0.05. The basic thought was to use Machine Learning models and more precisely CNN, so that we could train the model correctly and each time the end user inputs an image from a satellite source or a UAV source to indicate the related areas with salinity, the salinity value in the highlighted area and the confidence (accuracy) score. The followed process of the current idea is widely known as classification. As it is more than clear, we could not have a number of classes related to every decimal number of salinities, because we would have an enormous number of classes. That was the reason why we selected the solution of rounding the salinity number and represent each rounded number with a range, in order to have a logical number of classes. We used 8 different classes. **Table 16** shows which salinity belongs to which folder. We used data for training and for testing.

datetime	salinity	salinity rounded	FOLDER
2021-07-12	0.487	0.5	TRAIN
2021-07-17	0.783	0.8	TEST
2021-07-22	0.8165	0.8	TRAIN
2021-07-27	0.691	0.7	TRAIN
2021-08-01	0.8915	0.9	TRAIN
2021-08-16	0.977	1.0	TRAIN
2021-09-10	0.6825	0.7	TEST
2021-09-15	0.328	0.3	TRAIN
2021-09-20	0.297	0.3	TEST
2021-09-25	0.243	0.2	TRAIN
2022-07-07	0.434	0.4	TEST
2022-07-12	0.386	0.4	TRAIN

Table 16 The categorization of the UAV images, used as training data or testing data, as referenced to the "FOLDER" column. Also, it is observable the salinity as it is rounded.

CNN based object detection, has shown many advantages when using in remote sensing. For instance, someone observing a UAV/satellite image can easily see areas or things that include points of interest, and not use in-situ machinery that demands manual configurations and operation [83]. The well-known CNNs (Convolutional Neural Networks) are divided in the following two categories:

i) R-CNN (Region-based Convolutional Neural Networks), faster-RCNN and R-FCN (Region-based Fully Convolutional Networks), better known as two-stage algorithms
ii) YOLO (You Only Look Once), YOLOv3 (You Only Look Once - 3) and SSD (Single Shot Detector)

The CNN that was selected in order to assess the salinity was the SSD ResNet50 V1 FPN (Feature Pyramid Network) 640x640. Such type of models is appropriate for initialization when training takes place on new datasets. As proposed in the following research [84] the chosen CNN model supports better performance when there is need for real-time detection, against similar CNNs, for instance EfficientDet D1 640x640 or SSD MobileNet V1 FPN 640x640. **Figure 81** depicts the ML model used. We trained the ML algorithm in Google Colab through the use of a GPU accelerator.

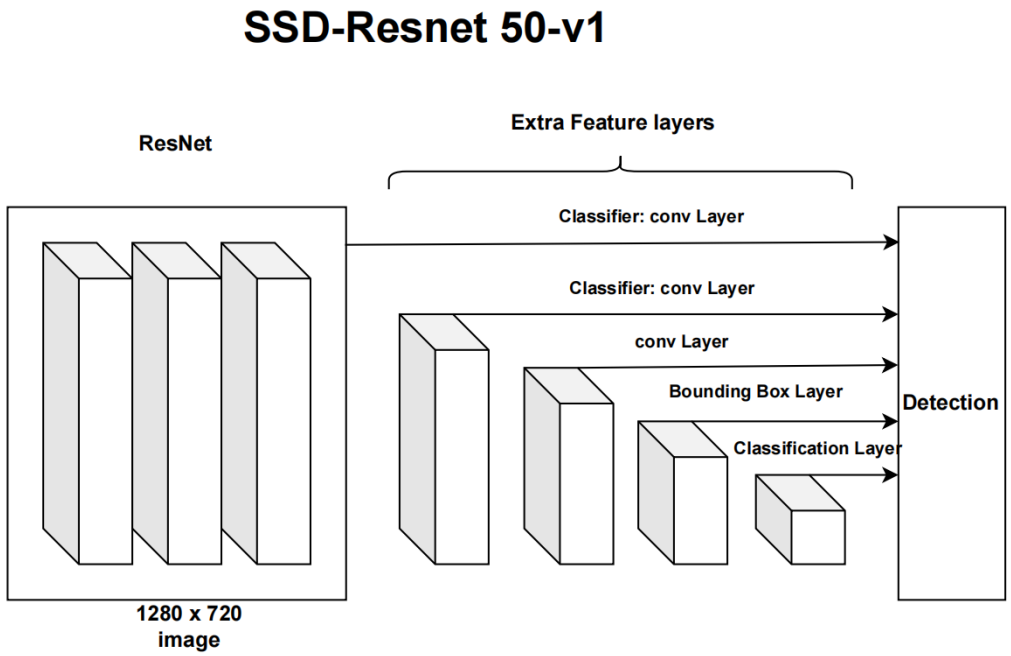


Figure 81 The SSD ResNet50 V1 FPN 640x640[37].

What we get after the object detection is presented in **Figure 82**. As it is more than clear, the images depict the covered area with the evaluated soil salinity and a percentage of confidence in %. So, the farmer can take the decision if it needs to open the water valves to irrigate their rice field or not.

[37] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

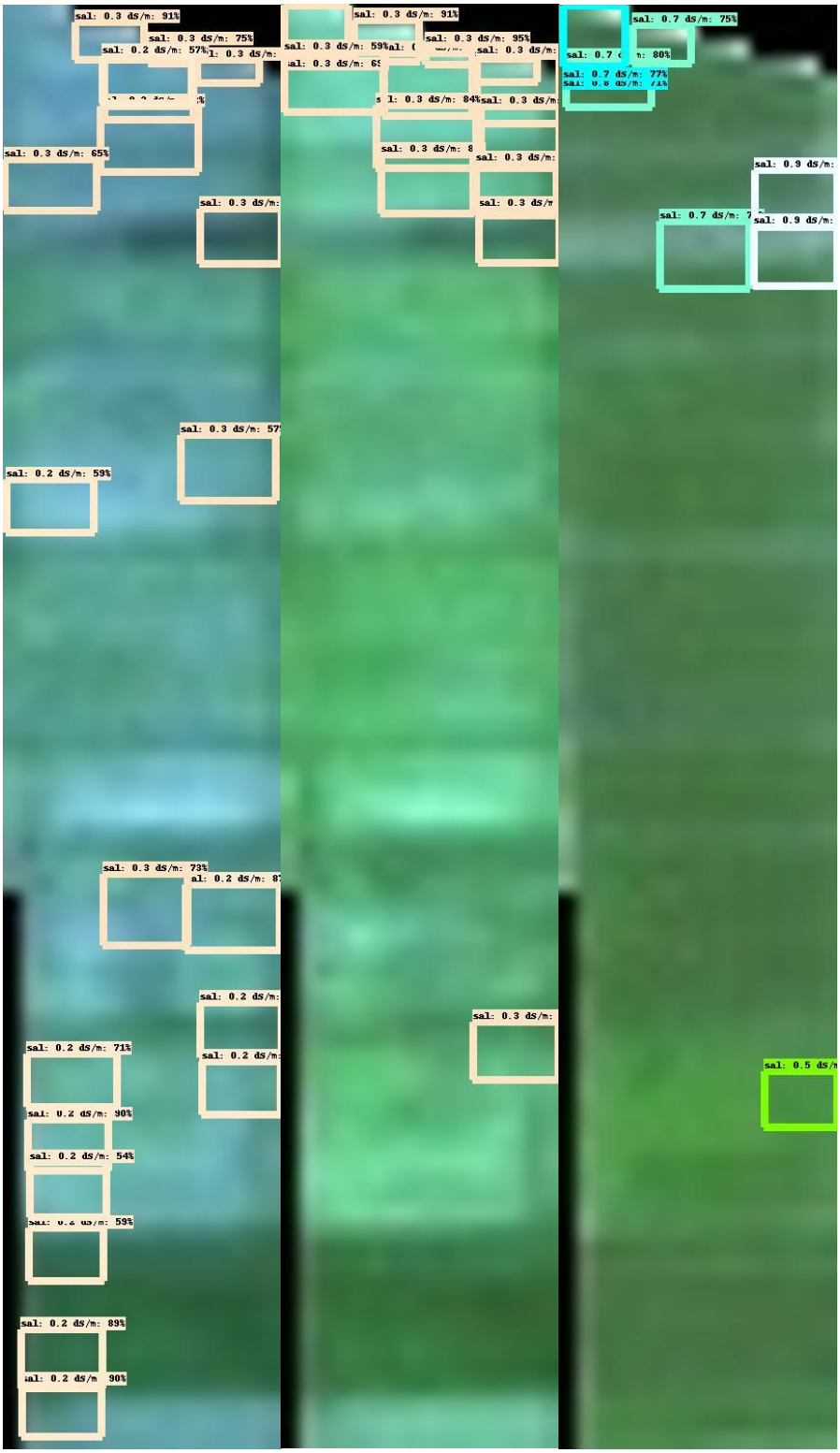


Figure 82 The object detection output images. As it seen there are rectangle showing the related soil salinity with the respected accuracy.

As an input we used the following image format: .tif file, 614 x 534 pixels, 32-bit float, 5 channels (Red, Green, Blue, RedEdge, NearInfrared. A typical example of the aforementioned format is given in **Figure 83**.

Figure 83 The images we used as input to the object detection model. Those were the initial images before the modifications in order to keep the area we need to work on.

In order to use the images, there was a pre-processing via the use of QGIS open-source platform for geo-imaging processing. So, we kept only 3 of the 5 bands, these were the Red, Green, Blue and removed the other 2 channels: RedEdge and NIR. There was a removal of the unnecessary farms in the image and only the ones with rice fields of the specific farmer were kept. The area that was needed to do the training and then the testing was extracted from the initial .tiff image through the use of a tool in QGIS, called shapefile. Then a python script was developed in order to enlarge the RoI (Region of Interest) with minimal distortion, aiming to feed the ML model.

Table 17 contains the parameter of the ML that was trained after feeding it with images, following the previously described formats and pre-processing. The number of classes was set to the

number 8, a number that could cover our needs, for the rice farm fields and the soil salinity. We chose Sigmoid function as a score converter. We used 2 different values for learning rate, one at the starting phase and one during the warmup rate, following the logic of divergent behavior avoidance. We selected 8000 steps, and the result showed that they were more than enough for the training. For training we were based on a ready-made model, the *ssd_resnet_50_v1_fpn_640x640_coco17_tpu-8*. The approach in cases of using ready models for classification, is to change the last stage with the classes to meet the requirements of the new training. As it was discussed before, we selected 8 classes.

parameters of the CNN	values
number of classes	8
score converter	Sigmoid
learning rate base	0.039
total steps	8000
warmup learning rate	0.013
warmup steps	2000
number of steps	8000
CNN base	ssd_resnet50_v1_fpn_640x640_coco17_tpu-8

Table 17 Parameters of the training.

Below, are depicted the various loss functions, and more precisely the following:

1. classification loss function (**Figure 84**)
2. regularization loss function (**Figure 85**)
3. localization loss function (**Figure 86**)
4. total loss function (**Figure 87**)

It is more than obvious that the loss is decreasing while the ML model continues its training, without signs of overfitting.

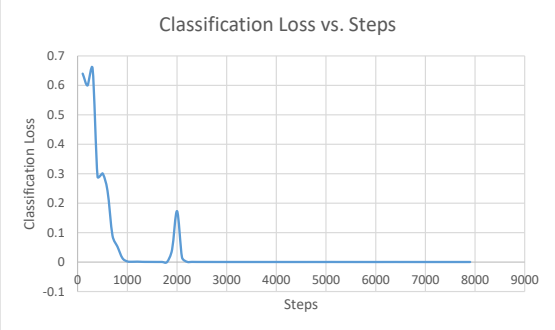


Figure 84 Classification Loss vs. Steps.

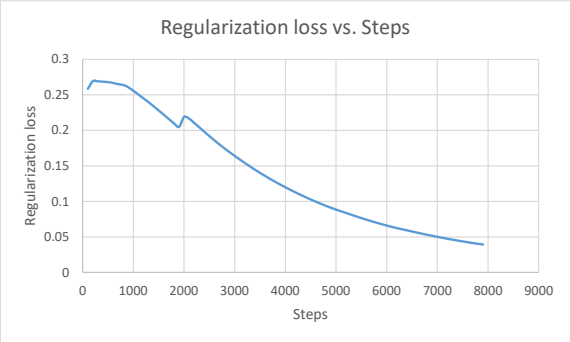


Figure 85 Regularization Loss vs. Steps.

Figure 86 Localization loss vs. Steps.

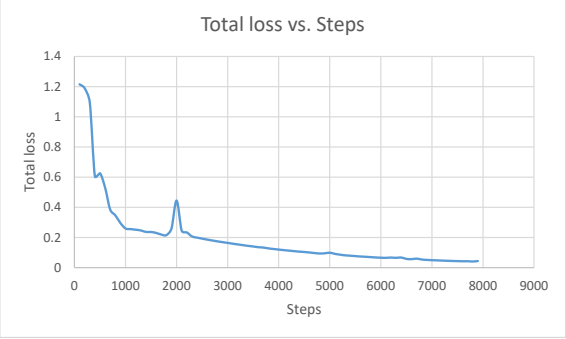


Figure 87 Total loss vs. Steps.

4.6 Conlusions

In the current chapter, a presentation of two Machine Learning models took place, with applications in the agriculture. The 1st one was the RNN-LTSM and its realization in Maize irrigation enabler. By using such kind of ML model, the end user can have predictions on various environmental parameters related to their farm field (local level), and not in general for a country. Thus, they are able to make decisions on when to irrigate their field or postpone the irrigation when too hot days approach. The 2nd representation was about a CNN model which was trained

on real satellite images and can estimate the soil salinity in Rice farm fields. We feed the CNN with satellite images and the algorithm can estimate the soil salinity outputting the areas and the percentage of confidence on the images. Thus, the end user is able to decide how much is the soil salinity in places on their farm field and decide if it worths to irrigate or not, saving in that way precious amount of water and money.

This page was intentionally left blank.

Chapter 5: An IoT device for monitoring resin/rubber collection from pine/rubber trees

5.1 Introduction

In pine trees, resin pipes are a solution in order to gather resin. Resin terpenes, as they are called scientifically, find implementation in many areas, to name a few: chemical, food, biofuel and pharmaceutical areas. The usage of resin production into selective cut pine tree cultivation happened by chance, as the standard tapping method in order to recognize high-resin yield, needs a lot of time and is very costly [85]. Resin includes turpentine and rosin. It is an essential chemical production coming from forests, and has too many applications. Pine resin has many uses (around 400) in areas in national economy, for instance in paper production, in the synthetic rubber, in electronics, in food, printing ink, oil paint, medicines and many other fields [85].

To give an estimation of numbers involved, during 2014, about 13.6 tons of resin was exported from India to China every year, making it the 10% of resin distribution in an international level. More than 40 countries buy resin from China, in a volume of more than 200.000 tons per year [86]. China was placed first in the world in the area of resin tapping related to pine forests, by exploiting 1.3 million hectares, producing around 6kg/tree/tapping/year, achieving 60.000 tons/year.

The extraction of resin from pine trees is called *tapping method*. In Indonesia they use the quarre technique. But this action, has many drawbacks. One could think that it has consequences in resin productivity. However, it goes a step further, since it affects the sustainability of the trees and the quality of the resin. An improved solution is the drilling method. **Figure 88** depicts the 2 most common methods for extracting pine resin from the pine trees. These are the quarre and drill methods

In [86] the researchers claim the following: About 32.64g per hole per tree, which consists the highest resin production was seen from the drill tapping technique. Around 19.34g per quarre per tree was identified in the quarre technique. In total, the farmers were able to achieve 9.29 tons/year via the quarre method and 15.64 tons/year via drill technique. According to another research [87], this kind of resin volume is considered economically profitable.

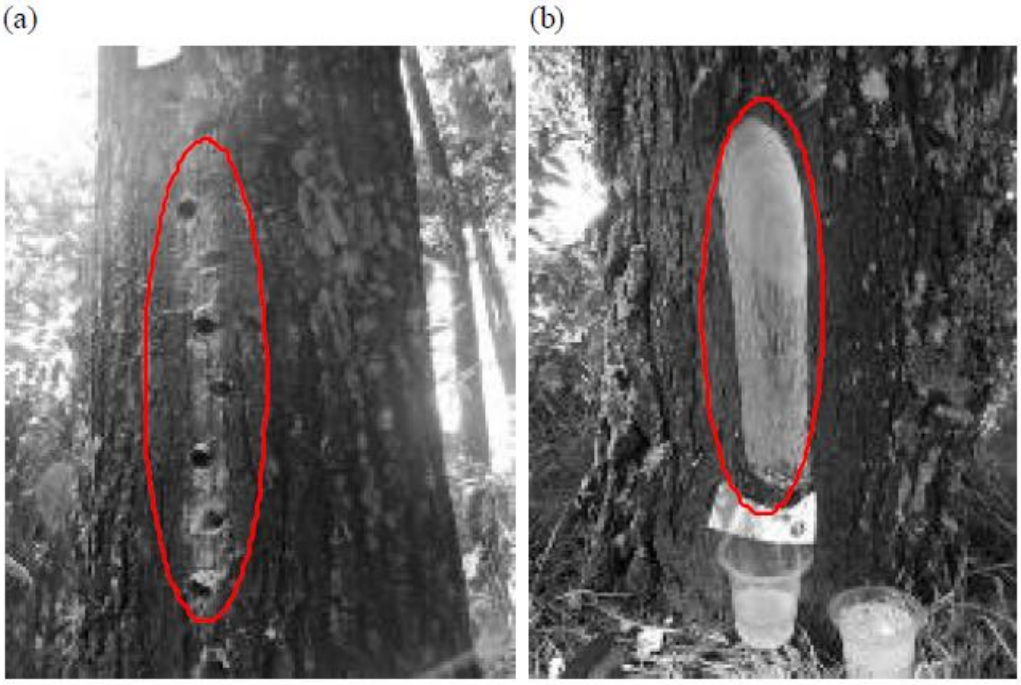


Figure 88 a) Drilled technique in a tapped Pinus merkussi, **b)** Quarre technique on the same tree [86].

The current chapter proposes a device for collecting resin for pine trees. It is an IoT device placed at the base of the tree in order to get the resin, coming either the farmer uses drill technique or quarre technique. The idea is that 2 buckets were used the one inside the other. Between them there are placed 4 load sensors. So, the resin that falls in the upper bucket is collected, and the load sensors weight it. The load sensors are connected to the Arduino microcontroller. The Arduino microcontroller is responsible to inform the end user either using Xbee Zigbee protocol or the GSM/GPRS modem with information about the resin collection. The following parameters can be data logged:

1) weight of the collected resin
2) environmental temperature/air humidity
3) weather condition, such as if there is rain falling or not
4) internal temperature and humidity condition inside the device

Thus, the farmer, is informed in near real-time about the resin collection and take measurements if something unwanted takes place. The device's parts are analyzed in a following section.

5.2 Existing literature

In [88] the authors propose a device, which is basically an industrial autonomous robot placed on an autonomous ground vehicle able to extract resin from pine trees and gather the oleoresin for further processing. Although there are industrial robots for manufacturing where there is control of between the robot's tool that operates and the targeted workpiece, in a scale of mm accuracy, when it is used in environments where there is high unstructured, such as forests or agri domain, there is a problem. The latter occurs because of the decreased accuracy existing in reality than

the increased targeted accuracy needed. Their research targeted on presenting the operations followed by the robot in order to drill 3 converging holes in a pine tree, spraying in those holes various chemicals and inserting a plastic tube. The difficulty of their mechanism was that they needed to accomplish their operation in external conditions (such as a forest) where there are many variations from tree to tree. They also described the many solutions they followed in order to achieve their goal.

In [89] the authors describe various resin extracting methods with traditional tools. As they support, the resin tapping takes place by exposing resin ducts via specific incision on the trees' stem. The greater number of natural resins are gathered in small quantities through forest habitants via the realization of traditional tapping methods. The current tapping implementations from chosen trees are the traditional used during the previous decades and they are specific for each location. Resin extraction is a difficult operation and attention is required when the specialized people are working in order to "pull" the resin out of the trees. The devices and tools that nowadays are used by the workers, incorporate injury to the trees and problems to the collectors consuming more time. In order to reduce drudgery of resin collectors and rise the efficiency in yield there should be materialization with small enhancement in the existing devices or techniques. For this reason, an enhancement in tapping techniques will diminish injuries to the used trees and guide towards a sustainable collection of resins.

In [90] the authors analyze the results of resin collection and identify fungal infection in resin, both tapped and non-tapped pine trees, using the as little as they can, invasive and non-invasive diagnostics. In pine trees the continuous harvesting of forest products, for instance resin extraction, is believed to have consequences to trees' vitality and as a result their affect to fungal diseases. This becomes a serious problem for standing vigorous trees, so the detection of those signals as early as possible is very important in order to design effective practices in order to control those kinds of issues. In their research the authors explore the effects of extracting resin, which was one of the important products in the Mediterranean area, as far as the pines' growth is concerned. They also detected and analyzed fungal presence in both resin-tapped and non-taped pine trees.

In [91] the authors investigate the effect of various extraction methods, such as hydro-distillation, microwave-assisted hydro-distillation, and solvent-free microwave extraction. They also investigated oleo-gum-resin types and various regions on yield, the activity of antioxidant essential oils gathered of the widely known Ferula persica gum-resin. The results indicate that oleo-gum-resins coming from Darb-e-Behesht generated more essential oil than oleo-gum resins coming from Sepidan by using each of the extraction methods described above. Every extraction method isolated more essential oil from Kokh samples than Shir's samples. The highest essential oil was reached by the solvent-free microwave approach from Kokh samples. They also collected 51 different compounds and as a result they identified the essential oil samples.

In [92] as the authors claim, the methods used for extracting gum, guide to reduction of gum yielding trees, such as Lannea coromandelica Hout Merr, from its natural environment. This guides to loss of wild germplasm from forests. As the authors state, they conducted one-year experiment in order to standardize the following:

i) tapping methods
ii) tapping seasons
iii) chemical concentration

on trees maintaining width of 80-150 cm in the Balodabazar forest, in the areas of Chhattisgarh. In order to gather maximum amount of gum from these trees, they used different tapping methods, without wounding the trees. The maximum gum production was gathered using Mechanical and Chemical tapping methods via the use of V shaped cut. As they claim, in mechanical tapping methods, maximum gum was gathered by square shape technique. They used Ethephon as catalyst in order to improve the metabolic activities in order to maximize the available gum in the gum channels.

5.3 Proposed device for collecting resin or rubber from trees

The current section analyzes the device that was designed and built in order to collect resin or rubber from resin trees or rubber trees and inform the user via GSM modem or Zigbee port of the collected resin as well as other parameters that are useful to the user of the device.

The resin/rubber collection device consists of the following parts:

1) Arduino MEGA 2560 R3[38]: This is (**Figure 89**) the "brain" of the device. The Arduino microcontroller contains the CPU with the related I/O ports which is capable of making all the computation of the device, such as reading the sensors, outputting information via GSM/GPRS modem, outputting information via Zigbee, open/close the relay in order to control the fan when the temperature is high inside the device, open/close the LCD, etc. Arduino MEGA 2560 R3 is a very famous open-source microcontroller solution. It operates at 5 Volts from the USB plug, or 7-12 Volt from the 2.1mm jack. The logic inside the board is 5 Volts logic. It incorporates 54 I/O pins (15 of which provide PWM (Pulse Width Modulation) output). It also contains 16 input pins. Its I/O pin can support up to 20 mA current, which is more than perfect for the resin/rubber collection device. Its CPU is clocked at 16 MHz. It contains 8 KB (Kilo Byte) SRAM, 4 KB EEPROM. As far as the connection protocols is concerned, it uses 4 UARTs (Universal Asynchronous Receiver Transceiver) ports, a I2C protocol and an SPI port. In the resin/rubber collection device we use the I2C port, and the SPI port to communicate with the rest modules.

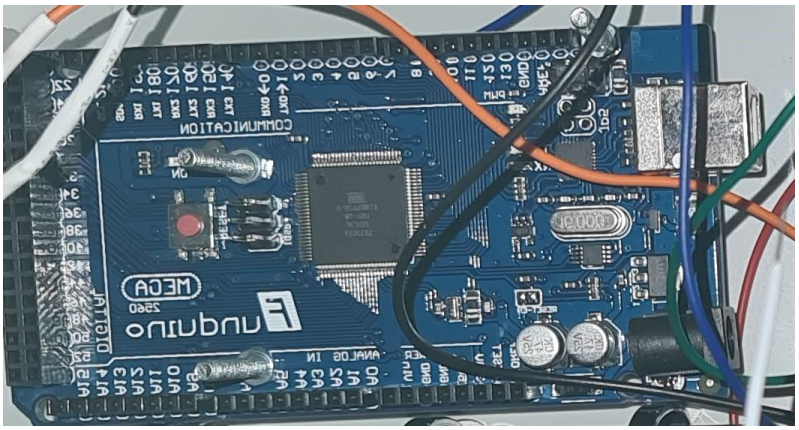


Figure 89 The Arduino microcontroller, which is responsible for supervising all the operations from/to various sensors.

[38] https://store.arduino.cc/products/arduino-mega-2560-rev3

2) Adafruit FONA[39]: This is the module (**Figure 90**) responsible for interacting with the user via GSM/GPRS network. The main purpose of this module is to send information of the whole "health" of the resin/rubber collection device when the user asks it via SMS. Moreover, it can be programmed to send information on specific time or day. The module operates on 5 Volts both supply and logic, and communicates with the Arduino MEGA 2560 R3 via a software Rx/Tx port. We use the module with the uFL connector. Along with the Adafruit FONA, we use a 1500 mAh LiPo battery, which is supplied by a mini-USB cable placed on the module. This battery is able to diminish the high current spikes up to 2Amps, when the FONA GSM module changes cell-tower.

Figure 90 The Adafruit FONA (GSM/GPRS) module.

3) Xbee Zigbee S2 2 mW adapter[40]: This is another module (**Figure 91**) to communicate either with other modules of the same type in order to exchange data, and construct an IoT network or with an end user via a Zigbee connected to a laptop. The idea is that more sensors can be assigned to the forest close to the resin/rubber collection device and send to the latter information that can then be provided to the end user. As a result, the resin/rubber main collection device can act as a coordinator. There is also the solution to change the S2 2mW Zigbee module with a more powerful that can reach 1 km coverage or even more and connect with the end user without the need to use GSM/GPRS modem. There are Xbee Zigbee modules such as the Xbee Zigbee 63mW Wire Antenna[41] that can reach 1600m in order to communicate. It operates at 3.3 Volt at 40 mA. It can send data in the rate of 250.000 bits/second. However, the one that the current proposed device uses, can reach 120m using 2 mWatt output (+3 dBm). It incorporates built-in antenna, 6xADC input pins, 128-bit AES encryption/decryption, as well as over-air configuration.

39 https://learn.adafruit.com/adafruit-fona-mini-gsm-gprs-cellular-phone-module?view=all

40 https://www.sparkfun.com/products/retired/10414

41 https://www.sparkfun.com/products/retired/10421

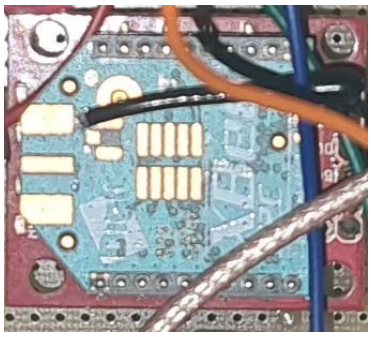


Figure 91 The Xbee Zigbee S2 2mWatt wireless module.

4) AM2315C – Encased I2C Temperature/Humidity Sensor[42]: This is temperature (**Figure 92**) and humidity sensor for external use, that can be placed inside the resin/rubber collection bucket without problem. It operates at 5 Volt both logic and supply and communicates with Arduino MEGA 2560 R3 microcontroller via I2C protocol. Its aim is to inform the user of the resin/rubber collection device with the outside temperature. It can sense humidity in the range 0 to 100 RH and temperature in the range -40 °C to 80 °C, with an overall accuracy +/- 3% for relative humidity and 0.5 °C for temperature.

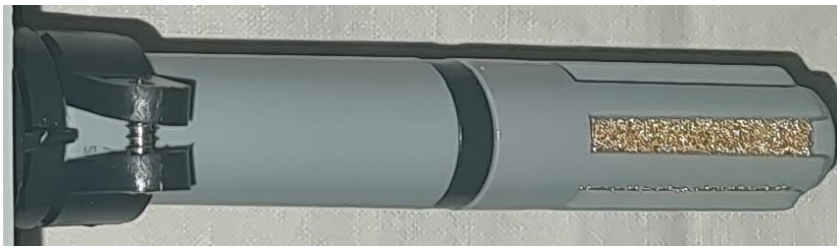


Figure 92 The external sensor for sensing temperature and air humidity in the environment.

5) Basic 16x2 Character LCD – White on Blue[43] : This is an LCD module (**Figure 93**), which consists of 16 columns and 2 rows, able to depict information in-situ when the user is physically close to the resin/rubber collection device by pressing a specific button. The module has in the back a potentiometer in order to adjust display contrast. It contains 4 pins, the following: VCC, GND, SDA, SCL. As it is clear, it uses I2C protocol to communicate with the Arduino. It operates on 5 Volts both supply and logic.

[42] https://www.adafruit.com/product/5182#description

[43] https://grobotronics.com/basic-16x2-character-lcd-white-on-blue-5v-i2c-protocol.html

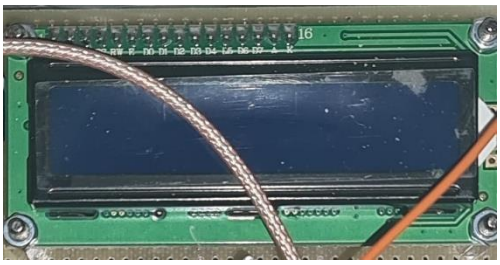


Figure 93 The LCD 16x2 screen, which is in charge to depict information to the user, via triggering the external push button.

6) Temperature-Humidity Sensor DHT22[44]: This is a sensor (**Figure 94**) for sensing temperature and air humidity placed inside the device's box. It is used for sensing the internal temperature of the device and when the temperature rises above certain threshold, the Arduino microcontroller enables the relay in order to operate the small fan and decrease the internal temperature. When the internal temperature decreases under the threshold, the Arduino microcontroller stops the fan via the relay. The sensor incorporates 3 pins: VCC, GND and OUT. It operates at 5 Volt both logic and supply. The consumed current is about 1.5 mA. It communicates with the Arduino MEGA 2560 R3 via digital pin (single wire). The module senses -40 °C to 80 °C temperature range with 0.1 °C resolution and +/- 0.5 °C accuracy. As far as humidity is concerned it can measure 0% RH to 99.9 %RH with 0.1% RH resolution and +/-2% RH accuracy.

Figure 94 The DHT22 air humidity/temperature sensor which is used in order to have a sense of the conditions inside the device.

7) SparkFun Load Cell Amplifier – HX711: This component (**Figure 95**) is used along with load cells and is responsible for reading the weighting of the resin/rubber. The aim of this module is to get the weight of the resin/rubber sensed via the load cells. The pins where the load cells are connected to HX711 are the following:

    a. Red (Excitation+ or VCC)
    b. Black (Excitation- or GND)
    c. White (Amplifier-, Signal- or Output-)
    d. Green (A+, S+ or O+)

[44] https://grobotronics.com/temperature-humidity-sensor-dht22.html

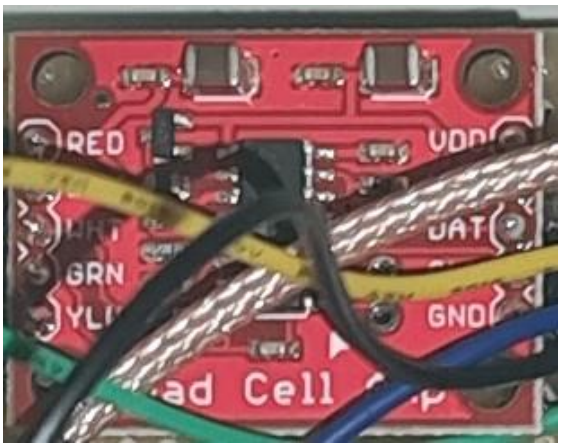


Figure 95 The Load Sensor Amplifier, which senses the resin weight in the bucket via the load sensors and sends the data to the Arduino microcontroller.

Below in **Figure 96** there is the schematic of the load sensors' circuit. As it is obvious the load sensors are connected to a Wheatstone bridge. Any difference in the load is sensed by one or more load sensors and their internal resist changes. As a result, the outcome, the end value, changes, something that is gathered by the Arduino microcontroller via the load cell amplifier.

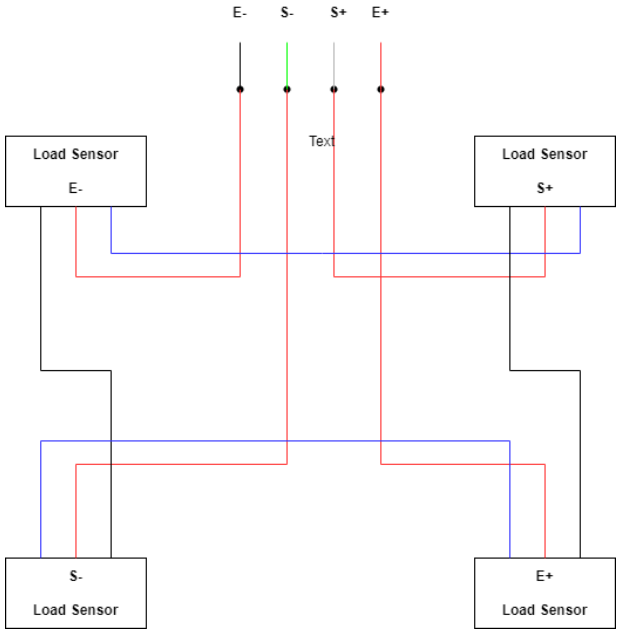


Figure 96 The circuit connection of the 4 load cells and the respected connections to the HX711. As it is obvious the circuit consists a wheatstone bridge.

Above in **Figure 96** there is the schematic of the sensors' circuit. They basically consist a wheatstone bridge, where each weight change is captured by the HX711 sensor module and transmitted to Arduino MEGA 25460 R3 via two cables, the DATA and the CLK. The module operates on 5 Volt both logic and power supply, with 1.5 mA current consumption. There is a choice of either 10 SPS or 80 SPS output data rate, where SPS stands for Samples Per Second.

8) Rain Sensor Module[45]: This module (**Figure 97**) sends a signal to the Arduino microcontroller when rain takes place. It uses the LM393 control board in order to transform the sensed rain into electrical signal. It can output both Analog or Digital signal. It operates at 5 Volt power supply and logic. So, when the rain starts and falls on the waterproof sensor, the module sends a signal to the Arduino controller which is informed about the rain.

Figure 97 The rain sensor enclosed into a rain-proof box. The rain sensor incorporates a 1-meter 2-wire cable, so that it can placed in the environment far from the central device. When water falls in the sensor, the Arduino controller instantly informs the end user.

9) Adafruit TCA9548A I2C Multiplexer[46]: This module (**Figure 99**) is a multiplexer 8x1 specialized in I2C protocol. So, if there are more than one I2C devices in the board, as occurs in the current resin/rubber collection device, this module can assign up to eight I2C modules in one I2C port. As it is obvious the multiplexer is connected to the Arduino's I2C port and can fan out eight I2C ports, so there can be access to up to 8 eight different I2C modules. In our case, the MUX (Multiplexer) connects the LCD 16x2 and the external temperature/humidity sensor to the Arduino microcontroller. The MUX uses 3 control pins (A0, A1, A2) in order to switch between $2^3 = 8$ devices with the following logic table in **Figure 98**:

[45] https://grobotronics.com/rain-sensor-module.html

[46] https://learn.adafruit.com/adafruit-tca9548a-1-to-8-i2c-multiplexer-breakout?view=all

Inputs			Outputs
A2	A1	A0	
L	L	L	Device-0
L	L	H	Device-1
L	H	L	Device-2
L	H	H	Device-3
H	L	L	Device-4
H	L	H	Device-5
H	H	L	Device-6
H	H	H	Device-7

Figure 98 Logic table of control pins and the related output (selected device)[47].

The TCA9548A operates at 5 Volt both power supply and logic.

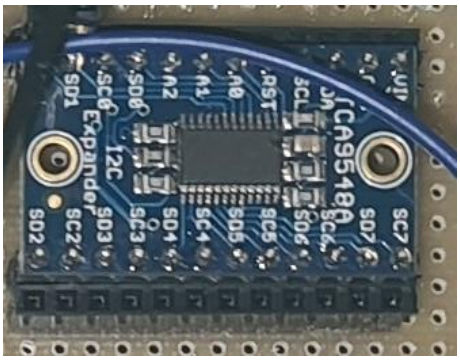


Figure 99 The TCA9548A I2C Multiplexer.

10) Relay Module – 2 Channel 5V[48]: This is the classical relay (**Figure 100**) that connects and disconnects loads such as a small fan or an LCD display, as occurs in the current resin/rubber collection device. The specific relay can operate on 5Volts DC, or 250Volts AC, at 10 Amperes maximum current. For the current device, of course, the current is going through the relay is less than 200 mA. The relay is controlled from Arduino's digital pins. We use 2 of them in order to control the two relays. We also use the VCC pin connected to 5 Volt DC power supply and the GND pin connected to the GND of the whole board.

[47] https://www.ti.com/lit/ds/symlink/tca9548a.pdf?ts=1698647991838

[48] https://grobotronics.com/relay-module-2-channel.html

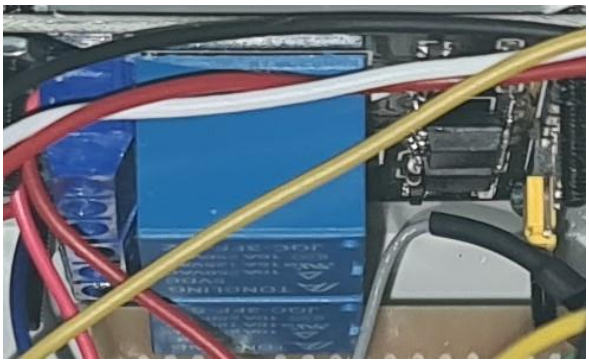


Figure 100 Two relay module, in charge to open/close the LCD screen and/or the fan.

11) Fan 5 Volt, 0.83Watt: This fan (**Figure 101**) is for cooling the internal part of the device and especially the battery which is connected to the solar panel. The resin collection takes place during summer season, when the temperature in Greece is high. So, there should be a cooling system for the battery and the electronics. The fan is connected to the 5 Volt power supply and controlled by the relay, which is controlled by the Arduino microcontroller.

Figure 101 The fan responsible to cool the whole circuit and the battery when there is increased temperature inside the box.

12) Fan guard[49]: There are 2 fan guards (**Figure 102**) placed in the 2 parallel sides of the box. Their role is to let the heat leave the box via the fan operation, and new cool air to enter the device. The fan guards also protect the device from objects accidentally enter the device.

[49] https://grobotronics.com/fan-guard-40x40mm-metal-silver.html

Figure 102 Fan shield.

13) Push button: This is the button (**Figure 103**), the user presses in order to get information about the sensors, when they are physically near the collection device. The push button is directly connected to one of the Arduino's I/O pin and the VCC, so it sends a High (5 Volt) signal. The Arduino then activates the LCD screen via the relay and depicts information related to the data and the device to the user.

Figure 103 Push-button, that the user uses when they want to get information in the LCD screen instantly.

14) On/Off switch: This is a switch (**Figure 104**) which is pressed in order to supply the protoboard and the Arduino with power supply coming from the solar panel and the battery. The supply voltage is 5 Volt/2 Amperes coming from the battery charger.

Figure 104 ON/OFF switch to start/close the device operation.

15) Green LED: It is used in order to indicate whether the protoboard and the Arduino microcontroller and the rest elements are supplied with current or not. It uses a resistor in order to mitigate the voltage in its pins, since a Green LED needs less than 5 Volt to operate. If no resistor is used, the LED would be burnt.

16) GSM Antenna SMA (SubMiniature version A) 2dBi 50mm[50]: This is a GSM antenna (**Figure 105**) connected via specialized RF (Radio Frequency) cable to the Adafruit FONA, in order to receive and transmit wireless signals to the GSM tower. It operates on the following frequencies: 1710MHz to 1880MHz, 1800MHz, 1900MHz, 1920MHz to 2170MHz, 824 to 894MHz, 880 to 960MHz. Without it there is no chance for the resin collection device to communicate with the end user either via SMS or GPRS.

Figure 105 GSM antenna of the GSM/GPRS module.

17) Waveshare Solar Power Manager (B) – 10000mAh[51]: This is the solar charger (**Figure 107**) which is connected directly to the solar panel in one end and the protoboard + Arduino microcontroller in the other end. What basically does is charging the 10.000 mAh, 3.7V rechargeable Li-po battery it contains when there is solar light outside and provides power supply to the resin/rubber collection device. The interface is depicted in **Figure 106**. In position [1] the solar panel is connected. In position [2] and [3] there are the 2 outputs, where the user can connect either a USB A plug or a TYPE-C USB. USB-OUT [3]

[50] https://grobotronics.com/gsm-antenna-sma-2dbi-50mm.html

[51] https://grobotronics.com/waveshare-solar-power-manager-b-10000mah.html

can support 5Volts/3 Amperes which is more than enough for the resin/rubber collection device. Position [4] is the battery on/off switch via which the user enables the power supply to the device. Positions [6] and [7] are the LEDs responsible for the condition of Solar charging with the following meaning[52]:

- Solar Warning: lights up if solar panel is reversed
- Solar Charge: lights up when recharging from solar panel
- Solar Done: lights up when the battery is fully recharged

Position [8] contains the battery life LEDs, depicting how much power-time is the battery capable of providing.

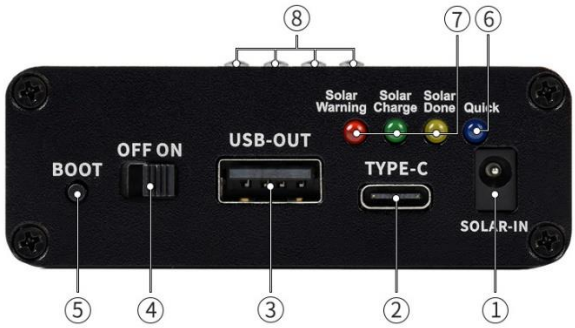


Figure 106 The solar power manager interface[47].

Figure 107 Solar power manager with coolers attached to every side.

18) Fuse Holder 5x20 with Wire[53]: This is the protection part (**Figure 108**) of the device when there is a short-circuit. The fuse holder contains a 600-mA fast glass fuse, which is burnt when a short circuit takes place. Thus, protecting the device from damages. A 600-mA fuse is used as a result of the calculation of the various modules power consumption used in the resin/rubber collection device.

[52] https://grobotronics.com/waveshare-solar-power-manager-b-10000mah.html

[53] https://grobotronics.com/5x20.html

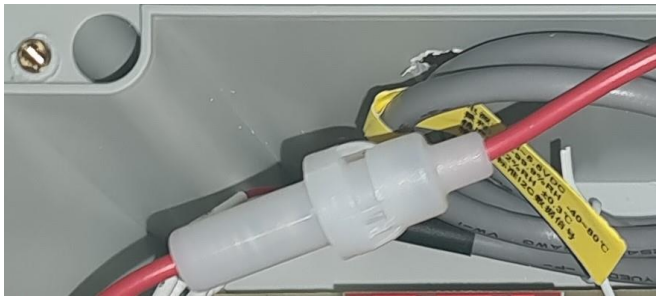


Figure 108 Fuse holder with a 600 mA fast fuse inside in order to protect the whole circuit from short circuits.

19) Load Sensors – 50 Kg[54]: These type of load sensors is capable of measuring up to 50 Kg. When a weight is placed on them, they change the value of the internal resistor they have, thus changing the value of the current. As it is depicted in **Figure 109** four load sensors are placed on a metallic disk. As it was presented in a previous paragraph

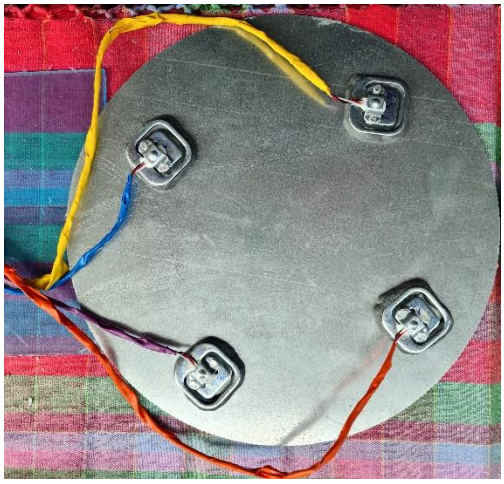


Figure 109 The lower metallic disc with the four load sensors attached to it.

the load sensors are connected in a wheatstone bridge. They are placed inside a bucket with the metallic disk and above them it is placed another metallic disk and another bucket. So, the idea is that when the resin is fall into the bucket the amount of resin/rubber is weighted by the Arduino microcontroller via the use of the load sensors.

20) Solar panel: Which is responsible for transforming the solar energy into electrical and charges the 10.000 mAh LiPO battery inside the solar charger. The solar panel (**Figure 110**) is the essential source of energy for the resin/rubber collection device, since the latter is placed next to pine or rubber threes in the forests. So, there is no external power supply from the network. Another power supply would be a small wind generator: however, the

[54] https://grobotronics.com/load-sensor-50kg.html

choice of solar panel is more than enough for our needs. The solar panel contains the following characteristics:

- Pm: 10 Watt
- Vmp: 18.0 Volt
- Imp: 0.55 Amperes
- Voc: 22.3 Volt
- Dimensions: 340*232*17mm
- Maximum Supply Voltage: 500 Volt
- Test Condition: AM1.5 1000W/m^2 25 ^oC

Figure 110 The solar panel used in order to supply with power the resin collection device.

In **Figure 111** and **Figure 112** someone can observe the overall device with all the components it incorporates. To start with on the left-hand side there are the 2 buckets, the one inside the other, between of which there are the 2 metallic disks with the four load sensors between them. Above the bucket, in the picture, there is the rain sensor which is connected with a 2-wire cable in the central device. In the middle of the image, someone can see the solar panel, which is responsible for supplying the whole device with power. And on the right-hand side there is the central device with the components, sensors, microcontroller, power manager and wireless transceivers in order to implement the whole process and transmit the data wireless to the end user.

Figure 111 Overall image of the pine resin collection device with the main device, the solar pane, the rain sensor and the bucket resin collection device.

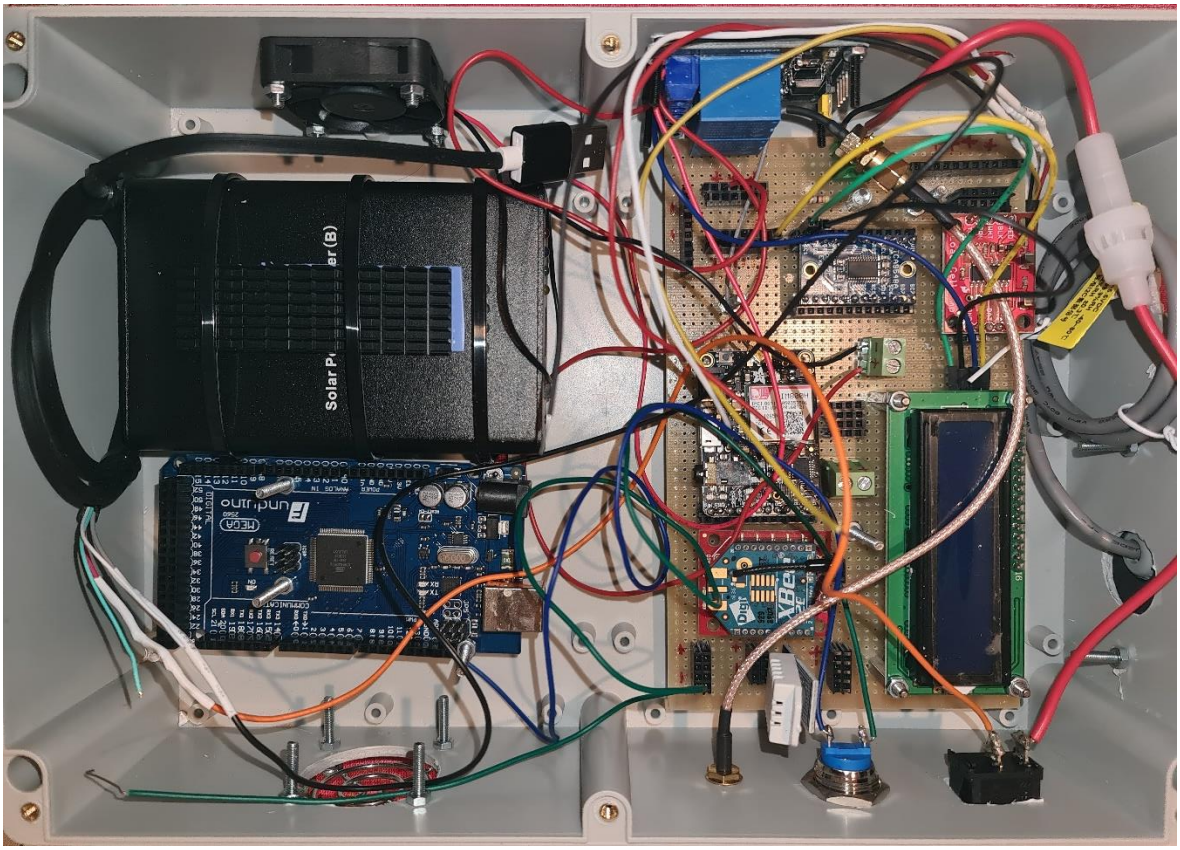


Figure 112 This image depicts the inside of the resin collector device, where all the elements and parts are seen. At the time of the screenshot the parts were not assembled in order to be clearly visible.

5.4 Results and Conclusion

In the current chapter there was an analysis of how the pine resin is collected from the pine trees and how significant the resin is for everyday life, since it has applications in many fields. Then, an extensive analysis of the literature took place regarding research in the specific domain. The essential part of the current chapter is the analysis of a modern/new device that can weights the resin collected from pine trees and inform the user, via Zigbee or SMS messages about the volume of resin collected in the bucket, as well as about the environmental conditions in the area that the device was placed. It is a very pioneering device, since at the time proposed (2023) there is nothing similar in the literature. The device could be used also for rubber collection since the idea of positioning it is similar. That's why throughout the word "resin", the word "rubber" is used.

Part 2: Security in Internet of Things

Chapter 6: Cryptographic protocols in Internet of Vehicles fields

6.1 Introduction

IoT technology can be implemented in transportation, where the integration of IoT shows intense challenges and opportunities in the commercial domain. Usage of IoT [93] [94] in the transportation area can guide to preventing accidents, improve issues such as: traffic congestion, traffic management, better scheduling. It can also improve wealth of drivers in fields like: intelligent smart aware navigation, automatic payment when entering tolls' area. By implementing IoT principles in vehicles there is the settlement of a network of vehicles and the idea of the famous IoV [95] [96], which stands for Internet of Vehicles, a very demanding area that contains the capability of impacting on people's lives.

In IoV there is extensive need for high security because it is involved with human lives. So, security against risks, internal, external, or both, is very essential. From this point of view, securing data coming from different stakeholders inside the IoV field becomes very crucial and demanding [97] [98] [99]. Furthermore, following the current directives related to data privacy, as occurs around the world, many researchers are examining methods in data privacy field, containing IoV idea.

Vehicles in the IoV domain exchange data each other and with the Roadside Units (RSU) by implementing the Vehicular Ad hoc Networks (VANETs) [100] [101]. VANETs have application in the intelligent transportation systems (ITS), easing the forecasting static wealth or dynamic wealth intelligence to the various stakeholders, such as: information related to: safety, street details, nodes, topology of frequently changing framework [102]. Lives of passengers or drivers can be harmed when altered data are sent on the VANET [103]. So, it is very crucial to implement data security and protection in data when transmitted or received over IoV networks [104] [105].

What the current chapter negotiates is the estimation of many protocols related to asymmetric encryption and decryption, for instance: ECC, RSA, NTRU, implemented in IoV topologies, by realizing different performance metrics, such as: during key generation process, generation of certificates, how long an encryption/decryption takes, signature times in generation or verification and other issues. AES cryptographic algorithm is the main scheme for all the measurements in addition to the different asymmetric protocols, named before. So, the different parameters measured, apart from asymmetric algorithm, were the size of messages, the pseudonym exchange, how new pseudonyms take place with their related energy consumption. The evaluation of the aforementioned was estimated in simulation via the use of ns-3 and SUMO open-source software, considering issues such as: CPU, power consumption, RAM in IoT environment, where they have constrained resources.

Research with privacy issues in IoV is very challenging [106] [107] due to the fact that passive and active attacks target on retrieving private information. When there is a passive attack going on, the attacker (intruder) analyzes the public data in order to identify sensitive information.

However, in the case of passive attack, the attacker aims on accessing private information in order to alter it. To give an example, when there is data poisoning attack, the attacker tries to input or alter regular data. This has effects in the performance of the training of the implemented Machine Learning algorithm, for instance a FDIA (False Data Injection Attacks) [108].

The essential idea of the current chapter is the analysis of an efficient method that maintains location privacy in an IoV network infrastructure. According to the evaluation findings, the method that current chapter delegates, has been modified in order to achieve better performance on metrics such as the size of messages, time duration consuming each message and energy consumption. The cryptographic algorithms used throughout the experiments are extensively analysed, also.

6.2 State of the Art related to IoV privacy

In [109] the authors propose a scheme which aware close vehicles and prevents attackers from detecting users through the use of Basic Safety Messages (BSMs), and at the same time decreases the transmission distance. So, the signal can extend only in the vehicles in the vicinity. This approach is named as WHISPER and is referred to road-safety due to the fact that many vehicles are in purpose blind to the tracker, but are observable to the nearby vehicles. The research is also related to many protocols and techniques used in order to form the distance of transmission and realize pseudonym transformation. WHISPER is estimated against known schemes in areas related to privacy contain, such as CPN (Cooperative Pseudonym Change), RSP (Random Silent Period) and SLOW (Silence at LOW speeds) through the use of a model that includes Manhattan-grid, with many densities, Quality of Service (QoS) and privacy of the location.

In [110] the researchers focus and update for attacks on VM (Virtual Machines) related to CE-IoV (Cloud-Enabled Internet of Vehicles) and analyze a new scheme that protects VM location privacy through the use of randomly selected VM identifiers. Their scheme takes advantage of the QoP (Quality of Privacy) idea by estimating the range of location privacy maintained, and at the same time they support their scheme enhances the QoP achieved.

In research [111], the authors proposed a scheme which is based on the concept of DPSZ, that stands for Dynamic Pseudonym Swap Zone, aiming at securing location privacy of the vehicles. In this scheme, every vehicle can induct a provisional pseudonym, by using the DPSZ. Such a vehicle can exchange its pseudonym with another random one that was chosen inside the zone. This operation is able to block DPSZ from exhibiting identities of the user the group's manager. Computation as well as communication overhead have to be decreased, and for that reason the scheme changes the pseudonyms in a way so that there is no linking between new and previous pseudonyms. DPSZ is able to fit to the quick transformations of the IoV environment targeting to reduce the cost of communication when there are many vehicles (nodes).

In [112] the authors proposed the Spatial Crowdsourcing (SC) concept for IoV, which uses decentralization of location while maintaining privacy. Their scheme is based on blockchain rationale in order to decrease the information produced by the IoV node/user through the SC server. It makes use of what is known as homographic encryption [113] and bears out the location of a vehicle via the use of circle-based principle, so that it can acquire the task's confidentiality taking into consideration the area's policies location. For managing user location privacy inside a

context with many levels, the scheme distributes a level of privacy for every user and displays this current on a mesh. Also, their proposed model takes advantage of the order-preserving encryption and zero-knowledge proof on non-interactive approach, thus preventing workers from illegally earning recompenses via the falsification of their driving locations.

In another research [114], the authors propose CSLPPS (Concerted Silence-based Location Privacy Preserving Scheme), which helps in IoV networks anonymity. It realizes unlikability in cases the users take part in location-based services and applications that have to do with safety for network vehicles. The proposed scheme settles a synchronization mechanism which operates when there is an identifier changing in silent period among the users before any information exchange with the new identifiers. The researchers implemented many simulations so that they could identify how their mechanism works including cases with GPAs (global passive attackers).

In [115] researchers introduce the TPPCD (Trajectory Privacy Preservation method based on Caching and Dummy locations), that targets on realizing location privacy protection by including false locations as well as caching in IoV infrastructure. In their scheme when the IoV nodes need LBS (location-based services), they show fault location in order to maintain their privacy. Moreover, RSUs, better known as Road-Side Units realize caching for decreasing the communication between the RSUs and the server which is responsible for the LBSs. The types of caching are the following two: “active cache update” and “passive cache update”. The active one has to do with the popularity and the passive one is related to false locations. Both active and passive schemes aim to bring location privacy and increased caching. The scheme proposed by the researchers can oppose LSAs (Long-term Statistical Attacks) and LCAs (Location Correlation Attacks), and at the same time can handle cache enormous hot rate.

In another research [116] the authors use the Paillier Cryptosystem in a modified version, in order to aggregate and transfer the data arriving from the vehicles’ sensors. The scheme they propose gathers information while operating as an IoV node for preserving privacy. It lets a vehicle produce structure from the collected information at various locations and make a report of mixed data that makes use of decreased resources both related to communication and computation parts. Their concept lets RSUs to aggregate data related to privacy-preserving data references and enumerates the number of reports of data belonging to each data dimension. The collected data are ready if someone needs to access them, moreover they can be re-encrypted and re-processed through the trusted management authority. Authors’ proposed concept identifies a node (data vehicle) that collects the data coming from the sensors concerning the remainder vehicles that belong to the IoV network. Management authorities do not involve, and at the same time RSUs can encrypt more the aggregated data to new encrypted texts and can be decrypted only from the data vehicles, without uncovering the real locations of the vehicles. The authors also claim the following for their scheme: a) it evaluates the possibility of data query problems, b) it estimates the level of security of the scheme concerning location privacy, c) analyzes the integrity and the durability in cases of collusion attack. Also, they support that their concept can mitigate in high volume the needs of various resources, such as computation and communication.

In [117] the authors propose a monitoring scheme able to maintain privacy in IoV and take advantage of the blockchain principle and reach transmission and obtainment between Mobility as a Service (MaaS) users. Their concept targets on backing privacy protection and vouching the collected data. Their scheme realizes a modified version of the famous Paillier cryptographic mechanism, moreover it implements identity signature in order to verify collection of records. It

is able to authenticate and aggregate historical data performance of any user. It takes advantage of the blockchains which contain PoS, better known as proof-of-stake unison for sending to others non-changed record related to performance and Bloom filter in order to store pseudonyms, targeting on prompting IDs of the transactions. The various pseudonyms belonging to the driver are made by the use of one-way hash functions and an identity-centered signature authenticates them. In order to have the historical performance of their mechanism, they use oblivious protocol. The positive results related to the computation efficiency and the resources used, are tested by various tests in performance [117]. The results that come out, show that their concept works better than the common traditional mechanisms as far as computation is concerned. Lastly, they collect information about efficiency and privacy trade-off so that they can monitor a correct number of records related to performance for every transaction.

6.3 Robustness and time complexity of the algorithmic schemes that were used

This section provides an analysis related to the robustness of the cryptographic schemes that were used, and more specifically the following four: RSA, ECC, NTRU and AES. The time complexity of each algorithm is very significant in order to succeed in the needed robustness.

RSA scheme is part of the Cryptography section better known as asymmetric cryptography, which differs from the symmetric cryptography, where the same key is used for both encrypting and decrypting a message. What asymmetric cryptographic schemes bring, is that they offer strong encryption which makes the decryption of the encrypted message very difficult and cannot be predicted by the attackers [118]. To enable adequate security, the key size should be bigger than 1024 bits, in order to make it difficult for an attacker to recognize the plaintext. [118]. The concept is that it is very difficult to factorize very long prime numbers [119]. Very large prime numbers, can process in the speed of computing, but it is impossible to human in Earth's time to factorize long prime numbers. The time complexity of RSA, given by the following paper [120] is $O((\log_2 x))^3$.

ECC scheme can output the same security as RSA or any system using discrete logarithms with significantly shorter operands, such as 160 to 256 bits against 1024 to 3072 bits. ECC schemes uses the well-known DLP (Discrete Logarithm Problem), so DL-schemes, for instance: Diffie-Hellman exchange of keys, could also be realized using elliptic curves [121]. The well-known NIST (National Institute of Standards and Technology) considers as safe the following elliptic curves: 2^{163}, 2^{233}, 2^{248}, 2^{409}, and 2^{571} [122]. ECC bases in finding the separate of random elliptic Curve, where Elliptic Curves connected to finite fields can provide a boundless source as a result of their affluent structure. ECC scheme is part of what is called 'one-way' scheme, where it is quite easy to go in the one way, but very difficult to go reverse. Although the encryptor chooses a graph in order to connect to the graph, there is no chance to find the solutions. The only way to find the solution, is by trying random numbers. In case someone wants to use brute-force techniques, research has shown that it is not a good technique, although there may be options for every bit size [119]. It is known also, that time complexity of ECC is $O(\sqrt{x})$ [120].

NTRU algorithmic scheme uses what is called as lattice-based SVP and is selected because of the increased security, speed and decreased complexity it can provide. NTRU incorporates more strength against RSA when a quantum attack takes place [123]. NTRU bases on the logic of smallest vector in lattices and its connected computations rely on tailless polynomial convolution

ring which maintains integer coefficient containing maximum *N* degree. To give an idea, let it have the following equation:

$$\mathrm{R} = \frac{Z[X]}{XN - 1}$$

where *Z* is a polynomial with variable *Z*, the R ring is relented up to *N-1*. The time complexity of NTRU is calculated to be *O(NlogN)* [124].

AES scheme was invented in order to provide security to government's areas. The AES algorithmic scheme uses as low as possible number of cipher blocks from a 128-bit input blocks and 3 types of key sizes, such as: 128-bit, 192-bit and 256-bit keys. The operation of encryption consists of 4 types, which are:

1) SubBytes
2) ShiftRows
3) MixColumns
4) AddRoundKey

When starting the encryption procedure, the input that was previously fed into the state will undergo a transformation such as the above 4 transformations. The AES computation is known as round function. The last round is different from the previous stages, because in the ending stage, the state does not undergo MixColumns transformation [125] as depicted in **Figure 113**.

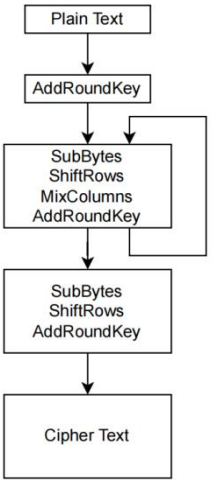


Figure 113 The AS operation as it depicted in a flow diagram.

The time and space complexity of AES scheme is *O(1)* [126]. In [127] the researchers include the **Table 18** which depicts different kinds of symmetric and asymmetric cryptographic schemes, having in common the security level.

Security Level	Symmetric key algorithms	RSA key size	ECC Curve
80	2TDEA	1024 bits	prime192v1
112	3TDEA	2048 bits	secp224r1
128	AES-128	3072 bits	secp256r1
192	AES-192	7680 bits	secp384r1

Table 18 Comparison of the different key sizes of RSA and ECC, having in common the same security level [127].

The authors of another research [119] contained in their paper the **Table 19** in which, it is one more time obvious the relation between RSA and ECC, concerning the same security level in bits. They also, make a reference in the ratio of ECC compared to RSA.

Security (bits)	DSA/RSA	ECC	ECC to RSA/DSA
80	1024	160-223	1 to 6
112	2048	224-255	1 to 9
128	3072	256-383	1 to 12
192	7680	384-511	1 to 20
256	15360	512+	1 to 30

Table 19 The current table shows different key sizes of RSA and ECC schemes, for the same security level with the related ratio of their key size [119].

In another research [123] the authors presented a comparison (**Table 20**) of ECC, RSA and NTRU protocols for the same security level.

Security Level	**RSA**	**ECC**	**NTRU**
80 bits	1024 bits		251 bits
112 bits	2048 bits		4411 bits
128 bits		256 bits	
192 bits		384 bits	5929 bits
256 bits			8173 bits

Table 20 The current table compares ECC, RSA and NTRU for the same security level [123].

6.3.1 Consequences of the key lengths in the security level

RSA and ECC use functions connected to number theory. What distinguishes them is the fact that they have to use long operands and keys. As it is obvious, a system is safer when the developer uses long keys and operands. So, in order to compare the different algorithms, the security level takes place. When we talk about a cryptographic algorithm which includes “security level of n bits”, it stands for that the attacker should make 2^n steps in order to penetrate the algorithm. This is something standard, as a result of the fact that symmetric algorithmic schemes which incorporate security level of n are connected to n-length bits.

The aforementioned are implemented only in symmetric schemes. When talk about asymmetric algorithmic schemes, there is a non-obvious relation between the algorithmic scheme and cryptographic security strength. **Table 21** presents symmetric and asymmetric algorithmic protocols for different security levels, such as: 80-bit, 128-bit, 192-bit and 256-bit, for different bit lengths [128]. As someone can understand, ECC uses much smaller key length than RSA or even DH (Diffie-Hellman)/DSA (Digital Signature Algorithm)/ElGamal to achieve the same security level.

Cryptographic Scheme		Security level (in bits)			
		80	128	192	256
Asymmetric Cryptographic Protocols	RSA	1024 bit	3072 bit	7680 bit	15360 bit
	Diffie–Hellman, DSA, ElGamal	1024 bit	3072 bit	7680 bit	15360 bit
	ECDH, ECDSA	160 bit	256 bit	384 bit	512 bit
Symmetric Cryptographic Protocols	AES, 3DES	80 bit	128 bit	192 bit	256 bit

Table 21 Symmetric and asymmetric schemes comparison, for different security levels [128].

6.4 Comparison between modern and classical asymmetric algorithmic schemes

In the current paragraph there is a comparison between the asymmetric schemes that were used, such as: RSA, ECC and NTRU with protocols published in the literature no older than 2-3 years. We start with the El Gamal asymmetric cryptographic scheme.

In **Table 22**, someone can see many asymmetric protocols up to the time writing the current PhD thesis (October 2023). In the first 2 rows, there exist 2 classical protocols, RSA and ECC. All the others belong to the family of Post-Quantum Cryptography. The table is organized as following: the 1st column depicts the protocol family, the 2nd column presents the variant of the protocol and the 3rd column proposes an estimation of each protocol against 2 unbreakable protocols, AES and SHA (Secure Hash Algorithms).

Protocol	Variant	Level of security
RSA	RSA-3072	1-At least as hard to break as AES128 (exhaustive key search)
	RSA-7680	3-At least as hard to break as AES192 (exhaustive key search)
	RSA-15360	5-At least as hard to break as AES256 (exhaustive key search)
ECC	P-256	1-At least as hard to break as AES128 (exhaustive key search)
	P-384	3-At least as hard to break as AES192 (exhaustive key search)
	P-521	5-At least as hard to break as AES256 (exhaustive key search)
	Curve25519	1-At least as hard to break as AES128 (exhaustive key search)
	Curve448	3-At least as hard to break as AES192 (exhaustive key search)
Saber	LightSaber	1-At least as hard to break as AES128 (exhaustive key search)
	Saber	3-At least as hard to break as AES192 (exhaustive key search)
	FireSaber	5-At least as hard to break as AES256 (exhaustive key search)
CRYSTALS-KYBER	Kyber512	1-At least as hard to break as AES128 (exhaustive key search)
	Kyber768	3-At least as hard to break as AES192 (exhaustive key search)
	Kyber1024	5-At least as hard to break as AES256 (exhaustive key search)
HQC	HQC-128	1-At least as hard to break as AES128 (exhaustive key search)
	HQC-192	3-At least as hard to break as AES192 (exhaustive key search)
	HQC-256	5-At least as hard to break as AES256 (exhaustive key search)
SIKE	SIDH-p434	1-At least as hard to break as AES128 (exhaustive key search)
	SIDH-p610	3-At least as hard to break as AES192 (exhaustive key search)
	SIDH-p751	5-At least as hard to break as AES256 (exhaustive key search)
CRYSTALS-DILITHIUM	Dilithium2	2-At least as hard to break as SHA256 (collision search)
	Dilithium3	3-At least as hard to break as AES192 (exhaustive key search)
	Dilithium5	5-At least as hard to break as AES256 (exhaustive key search)
FALCON	Falcon-512	1-At least as hard to break as AES128 (exhaustive key search)
	Falcon-1024	5-At least as hard to break as AES256 (exhaustive key search)
Rainbow	Rainbow-I-Classic	1-At least as hard to break as AES128 (exhaustive key search)
	Rainbow-III-Classic	3-At least as hard to break as AES192 (exhaustive key search)
	Rainbow-V-Classic	5-At least as hard to break as AES256 (exhaustive key search)
SPHINCS+	SPHINCS+-SHAKE256-128f-Robust	1-At least as hard to break as AES128 (exhaustive key search)
	SPHINCS+-SHAKE256-192f-Robust	3-At least as hard to break as AES192 (exhaustive key search)
	SPHINCS+-SHAKE256-256f-Robust	5-At least as hard to break as AES256 (exhaustive key search)

Table 22 Comparison of PQC algorithms against classical cryptographic schemes in order to give an estimation of difficulty to be penetrated [128].

From **Table 22**, two well-known cryptographic schemes were selected to be analysed in the following section, so that the reader get an idea of the referred mathematics and how they cause security by exceling the traditional rationale of algorithms such as RSA and ECC. So, in the following two paragraphs PKE (Public Key Encryption) and KEM (Key Encapsulation Mechanism) are analysed.

Attackers can penetrate classical algorithmic schemes such as RSA and ECC, via the use of large-scale quantum computers. It is very important that there are cryptographic algorithms that cannot be hacked via quantum computers. The latter are more widely known as PQC (Post Quantum Cryptography). The NIST has started to make standards some PQC algorithms, as well as the Hamming Quasi-Cyclic (HQC) which is based on code. Encryption based on code is based on error-correcting code, such as decoding vectors with small size random-based quasi-cyclic codes.

The KYBER protocol matches its security algorithm via modifications on the k parameter that can get 3 different values, such as: 2, 3, 4. Key production calculation needs *2k* NTT (Number Theoretic Transform) calculations and k^2 CWM computations. As far as the encryption is concerned, it needs *k* NTT, $k^2 + k$ CWM (Cluster-Weighted Modeling) and *k+1* INTT operations. Decryption, on the other hand, takes *k* NTT, *k* CWM and 1 INTT operations. CRYSTALS-KYBER consists a lattice-based

cryptography algorithm. Those types of cryptosystems work with polynomial rings and realize costly polynomial arithmetic, such as 2-polynomial multiplication with large-degree [129].

The current section was referred in different asymmetric protocols of the traditional algorithms, such as ECC, RSA, ElGamal and modern PQC algorithm such as Hamming Quasi-Cyclic and CRYSTALS-KYBER. Furthermore, PQC cryptographic schemes can resist against Quantum Computer attacks. A table was placed that presents the most modern PQC schemes connected to asymmetric encryption having as a "compass" classical unbreakable cryptographic protocol.

6.5 Proposed Scheme

The current section delegates a new scheme for maintaining location privacy in IoV networks and which is based on the lightweight MixGroup rationale [130]. Many messages that are sent inside the period of MixGroup operation include encryption in data so that there exists integrity of the data, IDs of the transmitted entities and protection against attackers (better known as eavesdroppers). The choice of the correct cryptographic protocol is very crucial, as somebody can understand, because it affects the efficiency of the model and it must conduce as much it can in order to enhance the security requirements set. There is need for fast response through devices with constraint resources in IoV networks, so, a serious mechanism should spend as low computation and power resources as possible, and at the same time minimize the size of data to undergo processing. In the proposed scheme the following parameters where considered:

1. Keys generation
2. Messages encryption/decryption
3. Production and verification of digital signatures
4. Production and verification of security verification

The Mix-Group idea was first appeared in this research paper [131] and is based on the following:

1. Not many vehicles are got together in global social spots, because the most of them are met in different places during their changed positions in the road network.
2. The majority of the vehicles includes individual social spots where they encounter many vehicles every day. Those positions maintain their stability in far time, especially the time of the day they visit them.

Having the aforementioned in mind, the social points are divided in two categories: global and personal. So, to manage maximum privacy as it can be, there is the necessity to make good use of the two keystones. The mechanism based to the Mix-Group rationale targets on both global and personal in the course of a vehicle so that it can build a coverage (place) where there is pseudonym exchange. There, a node (vehicle) can constantly exchange pseudonyms in order to reach the best secrecy it can for its identity. The nodes (vehicles) which are inside the area are transforming to member of a team and they make use of a common identity for the communication with the rest nodes and the related mechanism as long as they are change positions inside the targeted area. Thus, they make good use of both kinds of social positions, as they exist in the extended field and are verified points of pseudonyms exchange.

The mechanism the current chapter delegates is based on the steps described below:

1. Mix-Zone is used in an area and includes a number of global as well as personal social points.

2. Every vehicle entering this area, requests to be a member of an existing team and it is given a group identity.
3. They are also be given 2 identities: a temporary and an exchange identity, that they use during pseudonym exchange operation with the various vehicles of the alike team.
4. Each node (vehicle) that begins changing points inside the space, it finds the privacy gain it gets via the pseudonym exchange.
5. Under the eye of the temporary identities and nodes (vehicles) do the pseudonyms take place.
6. Every vehicle that leaves the team enables its fresh pseudonym and makes use of it for the future.

The current chapter's proposed mechanism provides enough protection in many cases of attacks. To start with, there is protection against depletion attacks when there is use of well-known standards of encryption and authentication, because it is impossible, from a computation perspective, to output information that someone can exploit only by possessing the requisite keys. Furthermore, by using digital signatures, an attacker cannot impersonate a valid node (vehicle) or change the message with an "evil" purpose. Thus, it is not easy for someone to forge an identity and the various messages related to RSUs. Due to the use of timestamps it is very difficult for a replay attack to launch. An attacker who is eavesdropping cannot systematically monitor a vehicle which comes in an area until it comes out of the area, because the adversary cannot associate the new pseudonym produced after vehicle leaves with one it used to have upon inserting the space.

When an internal adversary transmits messages containing not the actual position, targeting to produce mess and accidents in the scheme, due to the fact that every message is signed, it helps in locating the adversary and give him responsibilities. In the same tone, an attack shapes someone in the dependability of a vehicle, who duplicates the identity is blocked via the signing of the messages and also via the use of certificates exported by the RA (Register Authority). The mechanism can withstand internal and external attackers (adversaries). In this kind of attacks, the internal attacker after exchanging information with a node he moves the data he has exchanged to the external eavesdropper so that he is able to observe him. This can be avoided if the monitored node exchanges ones more pseudonym, something very common.

The proposed scheme enables alternations between the methods examined through the use of common logic variable: this occurs for the simulations and for having good construction as well as ease of use. To analyze the latter, each different cryptographic scheme is stored in a different file or library that includes the C/C++ functions that realize the aforementioned applications. In the following paragraph someone can see each method.

AES algorithm: was the "level 0", the base algorithms for each encrypted message, because it was selected in order to encrypt data for sending them, and after that the symmetric undergone an encryption with one of the messages studied that is analysed in a following paragraph. To further explain this: all the three cryptographic protocols used make good use of the AES for the initial encryption. The open-source library Crypto++[55] was used for realizing the AES algorithm, meeting the following parameters:

[55] https://www.cryptopp.com/

1. Operation method: Cipher Block Chaining (CBC)
2. 256 bits key size
3. 128 bits block size

Symmetric encryption is used in order to produce as random symmetric key sized at 256 bits, is initialized with an IV (initialization vector). Then, the text is transformed to hexadecimal-based format and is fed to the encryption algorithm. After the end of the operation, an encrypted text is the output in hexadecimal form. In order to realize the decryption, the procedure is inversed, in such a way that the text is changed from hexadecimal format to byte format, then the decryption of the symmetric key takes place, through the use of one of the public keys' cryptographic procedures, and finally the message is decrypted.

RSA algorithm: Similar to the AES protocol, the realization of RSA was based to the open-source Crypto++ library. The library is based in C++ and can be used to the NS-3 platform for simulations. The key size, as the only parameter configured throughout the simulations, was able to take three values: 1024, 2048 and 3072 bits.

ECC algorithm: The open-source library easy-ecc[56] was used for the implementation of the ECC protocol. The library is realized in C programming language and of course it is supported to the NS-3 simulation platform. It provides support for the following elliptic curves: secp128r1, secp192r1, secp256r1 and secp384r1. The implementation that the current chapter delegates, is the secp256r1 and offered 128bit level of security. Consequently, the private key size 256 bits (or 256/8 = 32 bytes), and because the library compress the representation, the public key size is 264 bytes or 264/8 = 33 bytes.

NTRU algorithm: The NTRU was implemented using NTRU-Crypto[57] open-source library. It is implemented in C programming language and it was used in NTRUEncrypt public key encryption protocol. The parameters used were according to EES449EP1, that can provide security of 128 bits with 623 bytes size of public key and 713 bytes size of private key. NTRUEncrypt protocol in order to work correctly needs an auxiliary secure, from cryptography perspective, random bit generator. For this purpose, the DRBG (deterministic random bit generator) was chosen, by using the file /dev/urandom[58], which everyone can locate in the Linux Operating System. The need is to initialize the generator only one time at the starting of each cryptographic operation. However, NTRU was not used for the realization of certifications related to public key and digital signatures. NTRUSign digital signature algorithm is not secure enough, as the possibility of leaking in private keys is significant. Although there have been proposed improved NTRU signature algorithms to be prototyped from the NIST, until the time of the current writing section, there have not been approved, that's why there have not been used to the experiments.

Realization of energy model: As far as energy in nodes are concerned, NS-3 supports the simulation of different models related to energy consumption and energy renewal[59]. The various models were implemented and significant information was output from the experiments. What the proposed scheme delegates is the measuring of the energy consumption when pseudonym

[56] https://github.com/arekinath/easy-ecc

[57] https://github.com/NTRUOpenSourceProject/ntru-crypto

[58] https://linux.die.net/man/4/urandom

[59] https://www.nsnam.org/docs/models/html/energy.html

exchange took place, so that there is evaluation of energy cost on the different protocols implemented. The model chosen for the experiments was the WIFI Radio Energy Model that can provide results concerning energy consumption of a node that can support WIFI protocols, moreover it was possible to simulate and support the following parameters:

1. Idle
2. Transmission (Tx)
3. Reception (Rx)
4. ChannelSwitch
5. CcaBusy
6. Sleep
7. Off

Every parameter is linked to a consumption value related to current (measured in Amperes) so the battery power left is easy calculated for the transition between the states. Also, there exists the choice to update the device's values when there is exhaustion of the power left, so the device can pause the exchange of messages, and in this way, it simulates very realistically the exhaustion of resources. So, the scheme is as following: every vehicle is linked to a source of energy, that maintains a random amount of available energy, however, enough to support the energy requirements of the simulation. When a message is exchanged in each vehicle there is a collection of difference between the old and current energy levels, and with this approach energy is calculated, for every message, in Joules. When a message is processed, function *update_entry_EnergyMap_Recv* renews the value of a fragmentation matrix, which holds the records for every vehicle with its related consumed energy for a certain type of message. The *energy_pivot* begins from a primary source value and get updated with each new message.

The realized scheme related to road network and vehicle traffic: The realization of road network and the related traffic due to nodes was based on SUMO platform. Following relative approaches[60] a 10x10 grid has been constructed, with each node being around 500 m distant from the rest neighbour nodes. All approach covered 10 km^2. From the 100 nodes, 40 were chosen to work as crossroads. Based on the latter, the end road network approach was built with aborting dead-ends and empty edges. The nodes selected by the use of *grid_generator* in order to produce random coordinates. Velocity was set at 19.45 m/s (equivalent to 70 km/h) maximum velocity, with the width of traffic lanes set at 3 meters, with each road consisted of 2 one-way lanes. The python script named *randomTrips.py* was responsible for the make of the vehicles in the network, that produces routes in a random way for random vehicles. SUMO got as input the files related to the topology of the network, including vehicles' routes. Then, SUMO, processed the files in order to start the simulation and built a file with the locations of every node during the execution. Via the use of the *traceExporter.py* script, the SUMO generated file was made to a file that could be used by the NS-3.

Network communication realization: NS-3 tool was used in order to handle the communication between the different nodes in the network, via the use C and C++ programming languages. Nodes such as RSUs and RA were first built. The vehicle nodes were counterpart with the input file made by SUMO, and RSUs and RA were placed in specific locations that could not change, because they were not dynamic. RA was placed at the centre position of the topology and the RSUs were placed

[60] https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement

to positions according to their number in counterbalance inside the grid. Nodes (vehicles) and RSUs used the WIFI 802.11p protocol to connect between them with the suitable IPs (10.1.10.XX subnetwork) whereas the RSUs and RAs linked to CSMA (Carrier Sense Multiple Access) to 192.168.0.X sub-network. The different stakeholders were communicating via UDP (User Datagram Protocol) packets, so it is easily understood that each node consisted of *send* and *receive* sockets. Regarding simulation, the rate of security messages has been 1 sec for the vehicles and the RSUs, while they were managed from callback functions used when a message was received. Special headers were used for the parting of the messages and the control of the realization of every control function. The headers used what they basically do is extending the class header of NS-3, so, the whole information is joined to the headers following a specific structure and order, as they are depicted in **Figure 114**.

Figure 114 The header message used in the simulation.

6.6 Evaluation of the experimental results

The current paragraph delegates the findings of the experiments in issues like efficiency of the cryptographic techniques used in IoV fields. For every calculation, counters were used in order to estimate time, via the use of *gettimeofday()* function. The function was triggered at the beginning and at the end of every calculation, so, it measured the between time, in order to output a sense of the time consumed for each calculation. Below, there are many diagrams that depict the mean values of every measurement for each vehicle, without taking into consideration the vehicles which did not participate in the exchange. Each calculation for the message produced during the phase of pseudonym exchange, follows the rationale presented in Section 3. Each vehicle consisted of two fragmentation matrices: the first recorded the whole size of each message and the other matrix carted the number of every type of message. Both matrices, contained the message type as the key, thus, making it easy at the end of every simulation to retrieve the mean message size for each different type. The same rationale, as above, was followed for measuring the energy consumption for every message. All the experiments[61] took place in Ubuntu Linux 14.0.4 Vitrtual Machine, running on Windows 10, with 4 GB RAM, containing 1-core Ryzen R7 2700X CPU clocked at 3.7 GHz. The software and tools used were the following:

1. NS-3, version 3.25
2. Netanim, version 3.107
3. SUMO, version 0.31.0

As depicted in **Figure 115**, the various experiments have indicated that considering the same level of security, NTRU cryptographic protocol needs about the 1/3 of the time of the ECC protocol needed for the generation of the key pair and about 1/400 of the time needed for RSA, a famous drawback in this area. In **Figure 116**, someone can view the time each protocol (*ECC-128, ECC-192, ECC-256, RSA-1024, RSA-2048, RSA-3072*) needs in order to generate certificate. NTRU was not used in these experiments because it hasn't been officially standardized for certificates generation. So, as it is seen, only 3 key sizes for RSA and 3 key sizes for ECC were used. **Figure 116**

[61] The datasets generated during and/or analysed during the current study can be accessed if needed after request to the author.

shows that RSA is intense affected from the size of key, whereas the ECC is affected in a smoother degree. More specifically, for security level equals to 128-bit, for key sizes of 256 and 3072 bits, both for ECC and RSA, the first needs 1/3 the time of RSA protocol for certification generation. Only common digital signatures used for certificates, so the well-known problem of RSA in signature generation is crystal clear.

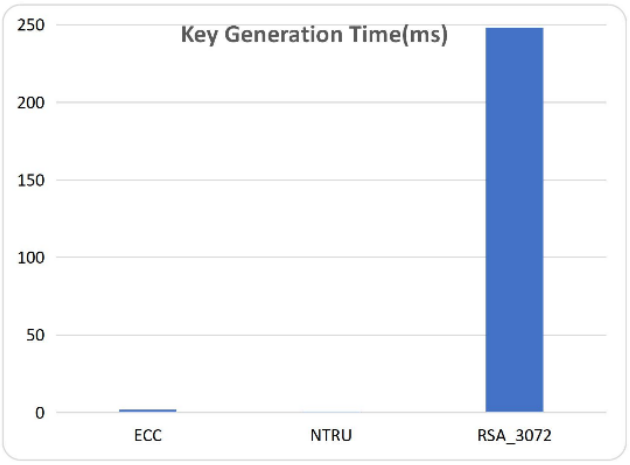


Figure 115 Time representation for the key generation, measured in ms.

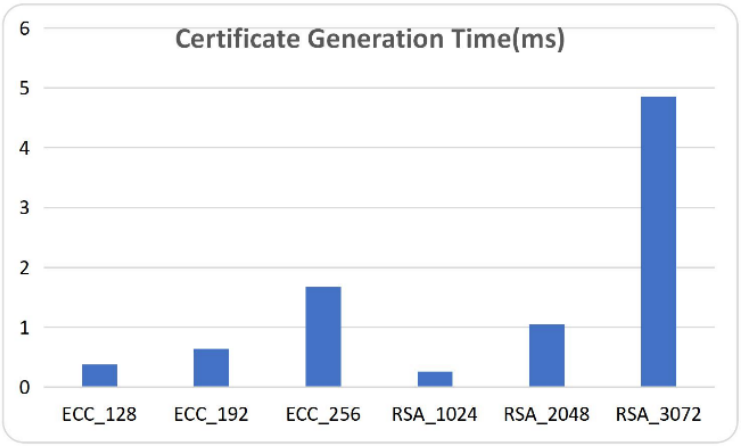


Figure 116 Time for certificate generation in ms.

As far as the encryption time is concerned for the three different protocols that are illustrated in **Figure 117**, the ECC algorithm is 10 times slower than the NTRU and RSA. This result can be explained because the two nodes compute their Diffie-Hellman secret (shared) each time they bring a connection between them, and not maintaining it static after the first time.

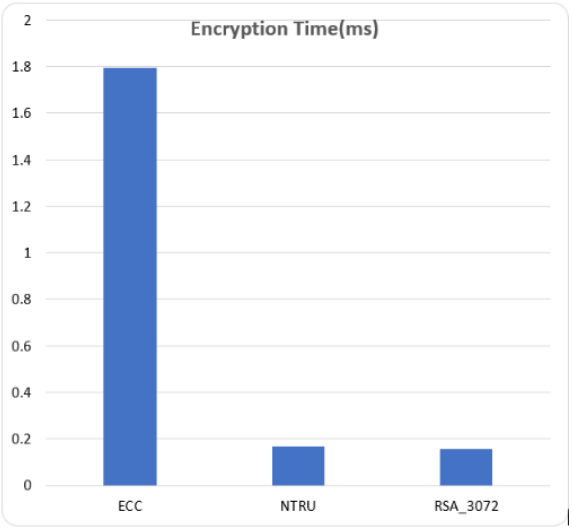


Figure 117 Time for the encryption phase, measured in ms.

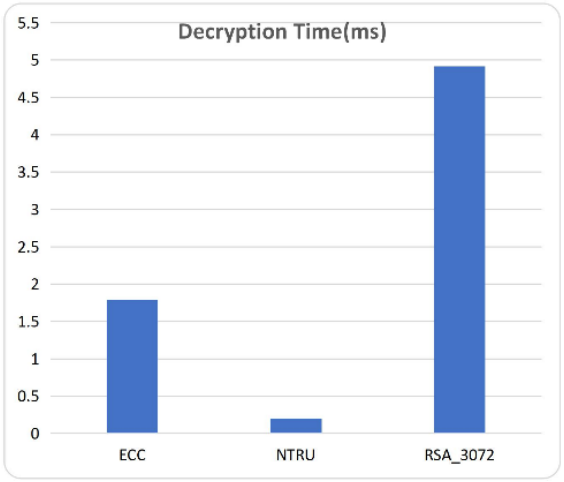


Figure 118 Time for the decryption phase, measured in ms.

Concerning the decryption procedure of the three algorithms, as they are depicted in **Figure 118**, the NTRU protocol seems to be the fastest of them, as it needs the 1/9 of the time of ECC and nearly the 1/25 of the time that RSA needs.

Another metric used in the experiments was the time of the signature generation. As someone can observe in **Figure 119** the whole picture of the results is similar to certificate generation with the same protocols (ECC, RSA) used. They differ in that the data used to output the results in **Figure 119** were coming from vehicles, whereas the certificates in **Figure 116** were coming from a RA (Register Authority) node. Concerning signature generation time, the ECC protocol is the most efficient. It confirms what it was presented theoretically in the literature, in a previous paragraph. On the other results, regarding signature verification time, the results are not in the same spirit. As depicted in **Figure 120**, the ECC protocol demands the 1/29 of the time of the RSA protocol for 128-bit security level (ECC-256/RSA-3072). It is obvious that the ECC is affected in a greater degree by the key size, comparing to the RSA: for instance, when doubling the key size,

the ECC signature time of verification is extended by 4.8 times, whereas the related time for the RSA extends by 3.6 times.

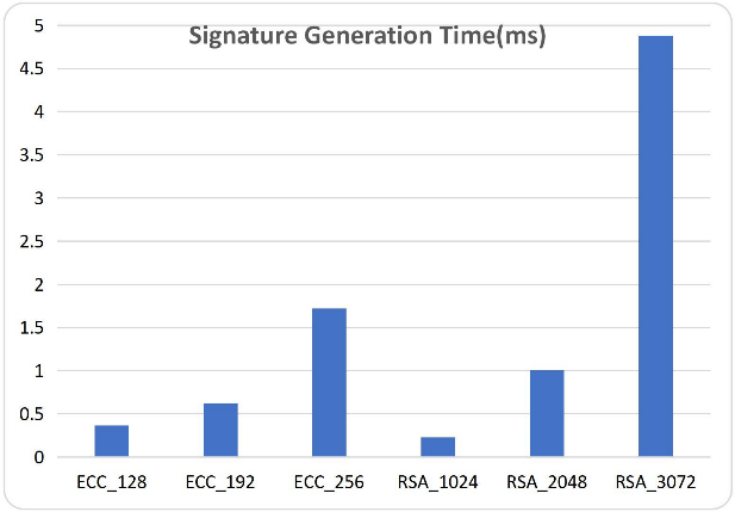


Figure 119 Time for signature generation, measured in ms.

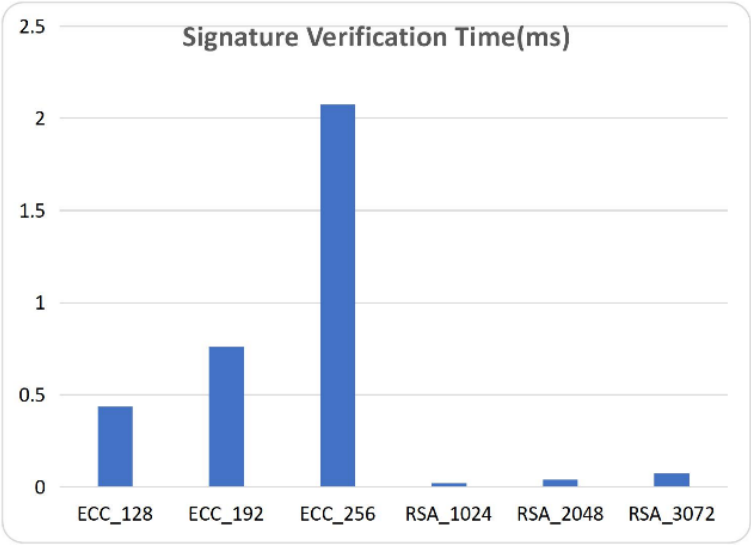


Figure 120 Time for signature verification, measured in ms.

In order to have a simulation as close to the reality, the following types of exchanges messages were used [131]:

1. *HEADER_TYPE_BROADCAST:* This is the message that each message sends periodically and maintains information about location, the signature and the certificate of its team.
2. *HEADER_TYPE_EXCHANGE_REQ:* It shows a request regarding pseudonym exchange, which a vehicle A sends in order to enhance its privacy. It is consisted of the temporary public key, sender's certificate, team's certificate and a timestamp.
3. *HEADER_TYPE_EXCHANGE_PROP:* The message that vehicle B transmits so that it can exchange pseudonyms with any random neighbour A. It includes the public key, the certificate after the encryption with the corresponding signature and the related timestamp.

4. *HEADER_TYPE_RESPONSE_CONF:* When the suggested pseudonym is exchanged, verified and is beneficial for vehicle A, then it sends a message including the public key of B and all the necessary data (public key, exchanged certificate of A) in the procedure of pseudonym exchange, including related signatures and timestamps.
5. *HEADER_TYPE_REPLY:* It is the response to the previous message (*HEADER_TYPE_RESPONSE_CONF)* and contains the related information from the vehicle B perspective, such as: key, exchange certificate, signatures, timestamp.
6. *HEADER_TYPE_PSEUDO_1:* It catches the information of vehicle A, such as identity, certificate and signature, which all of them are encrypted by using the public exchange key of vehicle B.
7. *HEADER_TYPE_PSEUDO_2:* It contains the data of vehicle B, which is encrypted using the public key of vehicle A, such as the first double signature (what data signature has vehicle A sent).
8. *HEADER_TYPE_PSEUDO_3:* It is related to the verification of the vehicle A. This vehicle sends the encrypted double signature and the corresponding timestamp.
9. *HEADER_TYPE_RECORD:* It is the message that vehicles A and B exchange. It contains the exchanged certificates for the two vehicles and the exchanged identities. RA's public key is used for the message encryption.
10. *HEADER_TYPE_RECORD_CONF:* It is a configuration message for vehicles' (node A and node B) records. They are the same.
11. *HEADER_TYPE_RSU_BROADCAST:* This message is transmitted by the RSU within a certain period, to all the nodes-vehicles in its coverage, containing its location and its public key.
12. *HEADER_TYPE_RSU_ACTIVATION:* The message transmitted by a vehicle to the closest RSU, so as to make an activation of its new pseudonym. It also contains all the exchanged information, which was encrypted by the public key of the RA.
13. *HEADER_TYPE_RA_ACTIVATION:* It is a message sent by the RSU to the RA. It is the same as this message: HEADER_TYPE_RSU_ACTIVATION.
14. *HEADER_TYPE_RA_NEW_KEYS:* It is the message with which the RA responds to a vehicle, when the latter makes a pseudonym activation request. It contains the id of the new vehicle, the two keys, and its certificate. All these are encrypted by the public key of the vehicle.

Figure 121, **Figure 122**, **Figure 123** depict the messages of every time, as discussed previously and based on the measurements that took place during the experiments, include the following cryptographic protocols: RSA, NTRA and ECC.

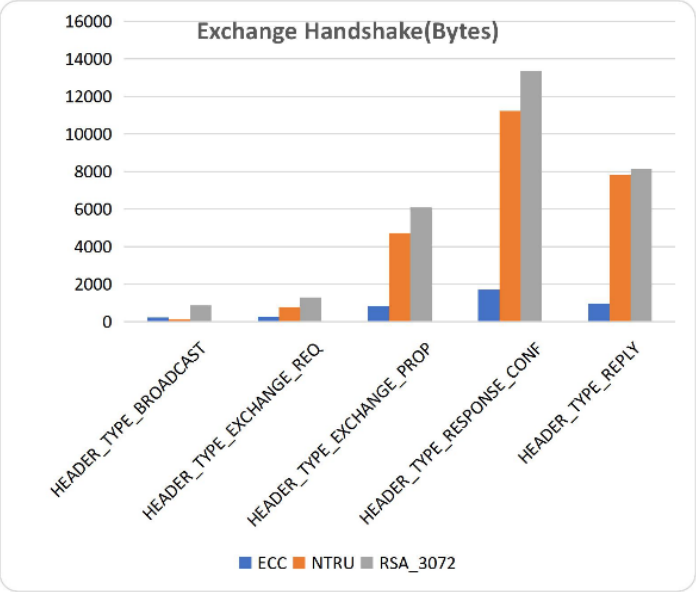


Figure 121 Size of messages, in bytes, during the negotiation phase.

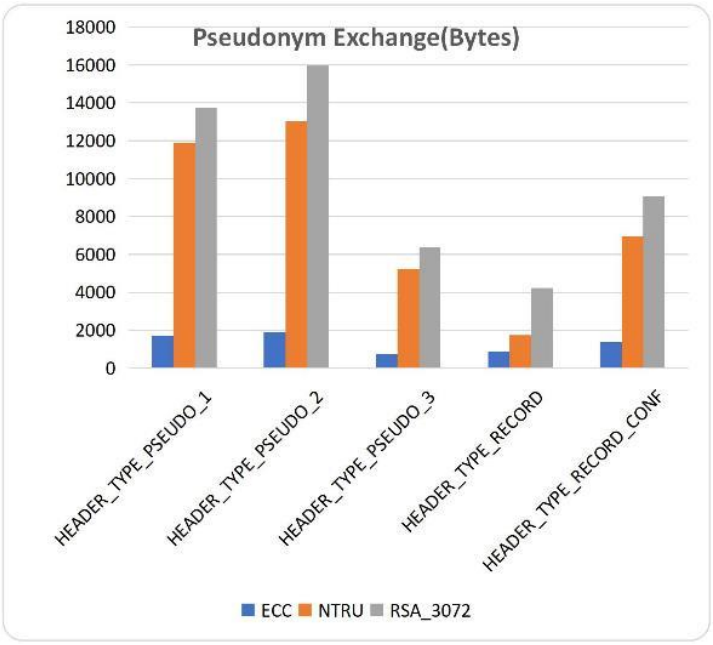


Figure 122 Size of messages, in bytes, during the pseudonym exchange period.

As presented in **Figure 120**, **Figure 121**, **Figure 122** the ECC protocol constructs messages with the smallest size. This was expected to occur, because those messages include keys and certificates. Large space covered in each message is due to the encrypted payload data. The message size is connected to the used algorithm and more specifically its public key size. ECC uses a public key size of 33 bytes, NTRU uses 623 bytes and RSA uses 384 bytes. So, it is obvious the reason why ECC and RSA differ. As far as NTRU is concerned the overhead of the digital signature was not taken into consideration, something that would offer extra increase of the size of the message in case the digital signature was taken into account.

Concerning energy consumption and energy levels, these were measured in Joules. The experiments that took place were based on pseudonym exchange period, due to the fact that during this phase they were more stable. During the negotiation phase the received or sent messages where varying in a great degree, because some of the nodes (vehicles) did not exchange messages, moreover, other nodes did not send requests to other nodes. The activation period is related to the constructed nodes, and due to the fact that nodes do not move in large nodes, the interest about energy consumption is very decreased. **Figure 124** depicts the energy consumption while pseudonym exchange was taking place. As it is obvious, the RSA protocol shows the highest consumption in energy, then follows the NTRU protocol and finally ECC protocol consumes the least possible energy. As someone can understand from the whole picture of **Figure 121**, **Figure 122**, **Figure 123**, **Figure 124** there is a link between the size the message occupies and the equivalent energy consumption. Thus, NTRU and RSA that contains large messages show increased energy consumption.

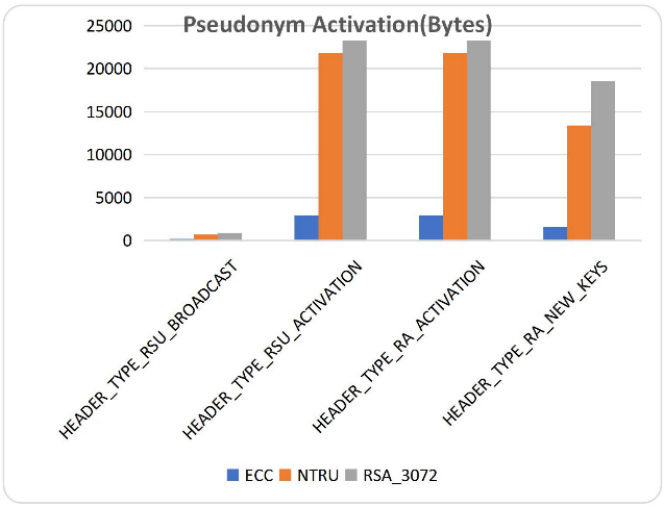


Figure 123 Size of messages, in bytes, during the phase of new pseudonyms enabling.

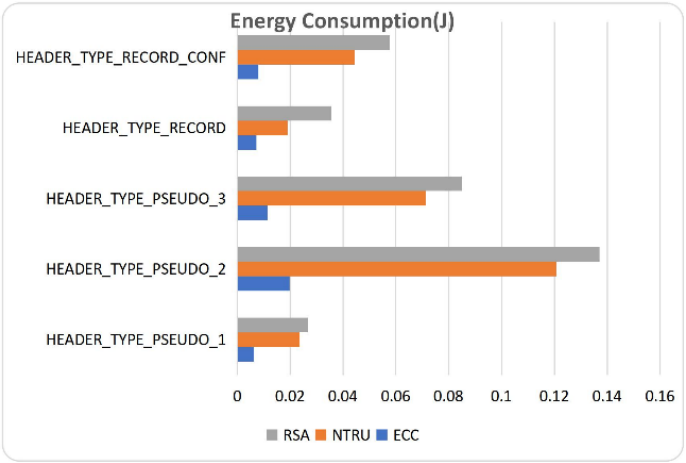


Figure 124 Consumed energy, measured in Joules, during the phase of exchanging pseudonyms.

Another experiment, the last one described in the current chapter, was consisted of a standard number of 5 attackers (internal adversaries) which stroked out the entropy of the nodes (vehicles). The vehicles were used in order to exchange pseudonyms. The body of the experiment was held in measuring 150 vehicles and frequency of measurement was every 30 seconds. Someone can observe that the entropy of the scheme increases very fast at the starting of the experiment and it then stabilizes. This happens due to the fact that as the time goes, even more vehicles become anonymous. The result output is presented in **Figure 125**.

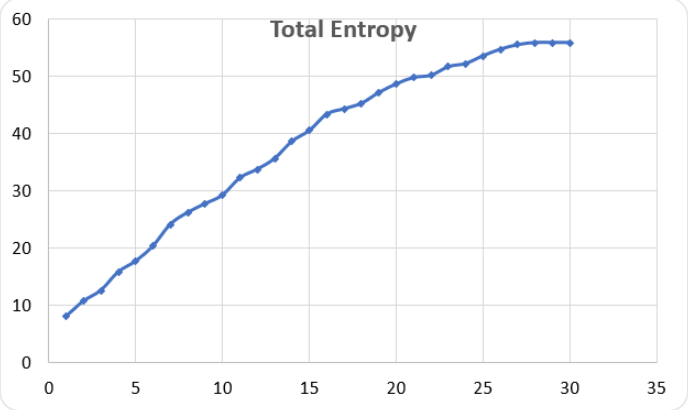


Figure 125 Total entropy of nodes (vehicles) throughout the time, measured in seconds.

6.7 The novelty that the current chapter provides in the literature

The solution that the current chapter provides is mainly based on algorithms like: RSA, ECC, NTRU and AES, which can be taken as the mainstream in securing vehicles. Furthermore, the ECC can make safe an IoV network consisting of many vehicles and RSUs, which need the least energy possible, extended to devices that operate with batteries, like cars, and vehicles that can be in the roads. The messages which maintain low sizes and the little time on many cryptographic computations give the ECC an advantage over other asymmetric protocols. **Table 23** depicts papers that used different solutions, but we do not know either what hardware the authors used or the various drawbacks that various protocols contained. As can be viewed from another point, RSA and ECC are known to be vulnerable to Quantum Computers attack. However, to mitigate this vulnerability we used NTRU cryptosystem which cannot be penetrated by those kinds of attacks. The latter did not mention by the authors in the literature presented in **Table 23**.

Paper	Proposed method	Advantages	Disadvantages
"WHISPER: A Location Privacy-Preserving Scheme Using Transmission Range Changing for Internet of Vehicles"	BSM (Basic Safety Messages)	A mechanism is proposed that informs nearby vehicles and blocks intruders from finding users via Basic Safety Messages (BSMs), while it lowers the transmission range so that the signal can reach only the vehicles in the vicinity	Since BSMs contain fine-grained location data, even though they are useful for road safety, they do open privacy-related issues: Any entity with eavesdropping capability can monitor the whereabouts of IoV users.
"Location Privacy Attacks and Defenses in Cloud-Enabled Internet of Vehicles"	CE-IoV (Cloud-Enabled Internet of Vehicles)	Proposes a scheme that guards VM location privacy via the use of random Virtual Machine identifiers. The proposed approach exploits the concept of QoP (Quality of Privacy) evaluating the level of location privacy preserved, while they claim that the proposed approach increases the QoP achieved.	An open problem exists, where there should be improved the QoP of schemes based on pseudonyms, moreover the repalcement pseudonym-based schemes, with schemes such as group-signature-based. Also, by replacing the VM's real identifiers with pseudonyms is not adequate for enabling well-preserved anonymity.
"Location privacy preserving scheme based on dynamic pseudonym swap zone for Internet of Vehicles"	Dynamic pseudonym swap zone (DPSZ)	In this scheme, each vehicle can settle a provisional pseudonym, by taking advantage of the DPSZ. The latter vehicle can permute its pseudonym with a randomly selected vehicle inside the zone.	(Nothing to mention)
"A Decentralized Location Privacy-Preserving Spatial Crowdsourcing for Internet of Vehicles"	Spatial Crowdsourcing (SC)	It is based on decentralization of location and privacy maintenance. The presented SC scheme uses blockchain technology to mitigate via the SC server the control of the data caused by the user of the vehicle.	Does not support privacy on task or solution content. Also it does not support real-world settings so as to manage difficult and complex real-world operations.
"Use of Homomorphic Encryption with GPS in Location Privacy"	Homographic Encryption (HE)	It uses homomorphic encryption and verifies the vehicle's location using a circle-based approach, in order to obtain confidentiality of the task concerning the policies of the area they are located in.	The problem with HE is that it is very slow.
"CSLPPS: Concerted Silence-Based Location Privacy Preserving Scheme for Internet of Vehicles"	Concerted Silence-based Location Privacy Preserving Scheme (CSLPPS)	It enables unlinkability when users participate in services based on location and also applications related to safety for networks of vehicles.	The average protection that it provides against an attacker is more than 75.6% but there is space to be improved. It does not guarantee 100% protection against a GPA.
"A Vehicle Trajectory Privacy Preservation Method Based on Caching and Dummy Locations in the Internet of Vehicles"	Trajectory Privacy Preservation method based on Caching and Dummy locations (TPPCD)	In the proposed approach, when users need to use location-based services (LBS), they expose dummy location for maintaining its privacy. Road-Side Units (RSUs) implement caching in order to mitigate the exchange of information among RSUs and the server that provides the LBSs.	It rises the overhead both for computation and communication tasks, affecting both the RSU and the LBS server.
"A privacy-preserving sensory data sharing scheme in Internet of Vehicles"	Modified Paillier Cryptosystem	It enables a vehicle to make a structure of data gathered at different locations and transform it to a composite data report that consumes reduced communication and computation resources. The proposed scheme allows RSUs to gather the privacy-preserving data reports and numbering how many reports of data are contained in every data dimension.	(Nothing to mention)
"Blockchain-Based Privacy-Preserving Driver Monitoring for MaaS in the Vehicular IoT"	Blockchain mechanism + MaaS (Mobility as a Service)	It can authenticate and accumulate someone's history of performance. Moreover, it exploits blockchains that contain proof-of-stake (PoS) unison for sharing unchanged performance record, as well as a Bloom filter to store pseudonyms, aiming for prompt transaction identification.	It does not prevent collusion attacks.

Table 23 Different proposed protocols are depicted based on different research papers targeting Location Privacy Protections in IoV environments [132][133][134][135][136][137][138][139][140].

6.8 Conclusions

Reading the gathered experimental results, interesting conclusions can be gained, as far as the optimization MixGroup is concerned, through its connection with the most appropriate encryption scheme. On the one side, the NTRU is able to give the quickest cryptographic key generation process for the level of security that the current chapter delegates. In IoV, the entity which is in charge for this operation is the RA (Register Authority). The latter produces these keys when the system is initialized. During the system procedure, there is frequently the necessity to produce new keys, due to the fact that a new vehicle is entering or maybe a set of keys have leaked, so, the operation should be very fast, making the NTRU, first priority. On the other side, concerning RSA, the experiments indicate the well-known advantages and disadvantages it encompasses. Once again, it is guaranteed that it does not work well when it generates keys, and at the same time it is the slowest of the three schemes when decrypting a message. Nevertheless, when it encrypts messages, it got ahead the other two schemes, and in a way balanced the previously described delays. ECC protocol is the most efficient when key generation takes place, and also is very fast in decryption and a bit slower in encryption, however faster than RSA by 30% in encryption and decryption mean time for 128-bit security level. As far as signatures are concerned, the ECC is a bit slower in validation signatures than signing, showing 25% less time than RSA for 128-bit security level. The fact that the specific implementation maintains slow signature validation is a disadvantage because the whole system demonstrates overload and delay.

The future plans are to extend the current chapter's work in order to contain the social aspect of the various vehicles which has consequences in the group structure, maintenance and progress. For that reason, an investigation could take place related to how the nature and the lifecycle of the different vehicle teams influence the performance of the aforementioned schemes and recognize advancements of the designed model. There could also be an investigation of the implementation of digital signatures through the usage of the NTRU protocol, thus acquiring a more integrated evaluation of performance. NIST was about to make standard the NTRU algorithm by July 2020, while implementations are through the 3rd round of assessment[62]. As soon as the NTRU is standardized, there will be an improvement related to NTRUEncrypt, with the aim to assess the related working out in end-to-end situations. Another issue could be the dealing with specific problems that were identified while the experiments took place, such as: synchronization of the vehicles, and loss of packets when collisions were taking place. A considerable future advancement could also be the settlement of an experimental network of vehicles in real conditions that would enable extensive testing of the previously described implementation in bigger topologies with the tart to enhance privacy protection schemes and urge the installation of secure IoV environments.

APPENDIX A

A.1. RSA pseudocode

The algortihm behaves as following [141]:

- First two prime numbers *a* and *b* are chosen.
- What is called modulus for the public, and private keys called *n* are multiplied by *b*
- An *e* should be selected which represents the public key and it is not a factor of $(\alpha - 1) \cdot (b - 1)$
- Then a private key *d* is computed and should be compliant with the following equation: $(d \cdot e) \bmod (\alpha - 1) \cdot (b - 1) = 1$
- The encryption is the output of the equation $C = M \cdot e \bmod n$ where *C* stands for the encrypted text and *M* represents the actual (original) text
- Then the decrypted is calculated by the following formula: $M = C \cdot d \bmod n$, where *C* stands for the encrypted text and *M* represents the original text.

A.2. El Gamal pseudocode

El Gamal cryptographic scheme [142] is close to the widely known Diffie-Hellman, where the common key *K* is produced based on long-term keys. So, for instance, if someone suppose that there are the 2 entities, the entity A, which acts for the sender and the B which acts for the receiver. The algorithm is as following:

$$A: y_a = g^{x_a} (mod\ p)$$

$$B: y_b = g^{x_b} (mod\ p)$$

62 https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement

$$A: K = (g^{x_a})^{x_b} (mod\ p)$$

$$B: K = (g^{x_b})^{x_a} (mod\ p)$$

In El Gamal cryptographic scheme, the long-term secret key x_α is superseded with an one-time key *k*, in order to be used directly for encryption.

To explain a bit more the various symbols:

$Z_p^* = (g)$: acts for the multiplicative group of the residue ring Z_p modulo *p*, where *p* substitutes a large prime number.

g: represents a cyclic group generator

x_b: acts for the long-term private key of user B

$y_b = g^{x_b} (mod\ p)$: represents the user B's public key

The steps of the encryption of the text *M* to be encrypted are as explained below:

A: $K = y_b^k\ (mod\ p)$, where k acts for the one-time private key

$$C_1 = g^k\ (mod\ p)$$

$$C_2 = K \cdot M\ (mod\ p)$$

$$C = C_1 || C_2$$

The decryption of the encrypted message C is obtained by the following formulas:

B: $K = C_1^{x_b}\ (mod\ p)$

$$M = \frac{C_2}{K}\ (mod\ p)$$

A.3. El Gamal modified pseudocode

The researchers in [142] propose an algorithmic scheme so that they can transform the classical ElGamal cryptographic scheme to an asymmetric cryptosystem. First of all, the researchers take as ground truth the fact that the value of *p* cannot get more than 512 bits. This canalize to the fact that solving the well-known DLP (Discrete Logarithm Problem) and acknowledging the long-term private key x_b can be succeeded in doing from a computational point of view for the chosen attacker. A serious finding from all open texts for many encrypted texts can be acquired from the following scheme:

$$M = \frac{C_2^{x_b}}{C_1}\ (mod\ p)$$

When a one-time private key is retrieved from the formula:

$$C_1 = g^k (mod\ p)$$

this enables the attacker to recognize a plaintext matched to the following equation:

$$M = \frac{C_2}{y_b^k}\ (mod\ p)$$

User B in order to confront those kinds of attacks, he generates *N* number of public long-term keys, such as: $y1_b, y2_b, \ldots, yN_b$, where number *N* acts for an extra security parameter. In order to let the one-time private key be more complicate, user A separates a part of the ciphertext C_1 in two parts:

$C_1 = C1_s||C1_r$, where the $C1_r$ is not transmitted.

Referring to the encryption of message *M* (plaintext), the next steps take place:

A: Choose by random the public key yJ_b from a pool of public keys: $\{y1_b, y2_b, \ldots, yN_b\}$ and produces the following:

$$K = yJ_b{}^k \ (mod\ p)$$

Where the *k* acts for the one-time private key.

$$C_1 = g^k (mod\ p)$$

$$C_2 = K \cdot M \ (mod\ p)$$

$C = C_1||C_2$ – ciphertext (encrypted text)

where: $C_1 = C1_s||C1_r$ and part $C1_r$ is not transmitted. As far as the decryption phase of ciphertext C the following occurs:

B: enumerates the not identified part $C1_r$, and yi_b for i resulting from the range: *{1,…,N}*

$$M = (C1_s||C1_r)^{xi_b} \ (mod\ p)$$

The idea if the choice of $C1_r$ and yi_b is correct or not is grounded on the "mechanism" of the plaintext. To give an idea, let's suppose that the plaintext can contain its hash *h(M)* for a few hash-functions that can withstand collisions. The B user should proceed through non-identified choices, and, for every choice to realize one exponentiation of polynomial complexity.

A.4. ECC pseudocode

Below are described the various parts of the ECC cryptographic algorithm [123].

Key generation: having in mind the Elliptic Curve *E* and the points *P* and *Q* on the elliptic curve, *n* acts for the maximum limit, so a *C* should be selected from the range *n[(1-(n-1))]*. As a result, the Public key is the outcome of the following formula: $G = c \cdot Q$ where the Private key is *c*.

Encryption: the sender transmits the message displayed on the curve. He chooses by random *j* in the range *1* to *(n-1)*.

$C_1 = j \cdot Q$ and $C_2 = M + j \cdot G$

Decryption: it is implemented by the following formula: $M = C_2 - c \cdot C_1$, with $C_2 - c \cdot C_1 = (M + j \cdot G) - c \cdot (j \cdot Q) = M + j \cdot c \cdot Q - c \cdot j \cdot Q = M$

A.5. HQC pseudocode

In this sub-section the quite new HQC (Hamming Quasi-Cyclic) cryptographic scheme is analyzed [143]. HQC cannot be "assigned" under Chosen Ciphertext Attack (IND-CCA). The protocol is built, grounded on the difficulty of a decision version that takes place on syndrome decoding on structured codes. It makes use of the *C [n, k]* code that can be decoded and a random double-circulant *[2n, n]* which describes an accurate upper bound for the analysis related to the upper bound.

This page was intentionally left blank.

Chapter 7: Use of cryptographic techniques of Hyperelliptic and Elliptic Curves in order to improve Privacy in Internet of Vehicles

7.1 Introduction

The Internet of Things, is a network which contains physical devices, vehicles or other elements which carry sensors or software in order to send their information, data to the Internet. A subcategory of IoT is the IoV (Internet of Vehicles), which enables the cares and the rest road infrastructure to connect with the Internet and send their information to many accepters. The V2V communication between the cars and between the cars and the rest machines such as the infrastructure (V2I) is very demotic nowadays. Via the strong evolution that describes the IoT and its possibilities it maintains to the enhancement of the experience which users face in a daily basis, the consideration in IoV passes off correspondingly. The modern technology guarantees automation, efficient utilization and convenience in the field of transportation, with the ultimate importance to be the bail of drivers' safety and the mitigation of the number of car accidents in the roads that results in increased percentage of people deaths nowadays. In the same tempo, as the 5G and 6G technology evolve, they offer very fast internet speeds and can enable a revolution in the area of IoV.

Although there is a rapid evolution which causes new dangers in the privacy of the users, there are new ways targeting to pervade users' personal information and take advantage of them. More specifically in the IoV field of communication (V2V/V2I) the security needs to be carefully monitored, because the intruders, apart from stealing private data, they can also put into danger the lives of the drivers. So, for that reason, in the current years there is power of establishing safer communication environment, which has to be adjusted to the constrained environment that the preconcerted machine provide when communicating. So, the security solutions of that special environments and in general when relating to IoT, it has to be efficient, quick and not promote the network with useless data traffic.

The whole research analyzed in the current chapter is related to privacy preservation schemes for VANETs, via the use of simulation environment in order to output detailed assessment. More details on IoV privacy clues, related research challenges and modern approaches can be found in [144][145][146][147].

The current chapter evaluates the exploitation of HECC in IoT machines in order to amplify the protection of privacy. The related approach is used and tested in IoV areas, where HECC signifies such a risen security by using small key. Moreover, we analyze speed encryption and decryption operations. In order to assess the current approach, many experiments took place using ns-3 open-source software, where the results are satisfying, when trying to enhance privacy

preservation in IoV. Many metrics were gathered, so that we could allow a recognition of the advantages and the drawbacks of HECC towards the protection of IoV privacy.

Last but not least, the current Chapter demonstrates a system realized in order to secure messages exchanged over the wireless network, and more especially, through the use of Xbee Zigbee modules, using the cryptographic protocols discussed above. The aim was to build, test and assess a system which makes use of commercial units such as the widely known open-source Arduino module and few low-cost electronics, for instance a voltage translator and a wireless Zigbee unit which lets users to exchange messages over the ISM (Industrial Scientific Medical) free channel.

7.2 Related State of the Art

The current section targets on providing an overall review of the most modern work focusing on privacy in IoV environments. The sub-sections below are organized as follows: the first subsection discusses on authentications mechanisms in IoV, the second sub-section analyses the secure message exchange in IoV, the third subsection demonstrates secure schemes that share the computational load on the network and the last subsection analyzes blockchain schemes exploited for privacy preservation in IoV.

7.2.1 Presentation of the authentication schemes

Via the uninterrupted evolution of IoT and IoV, there were many algorithmic schemes that were proposed in order to obtain secure and efficient communication in VANET infrastructure. Some of those schemes are based on authenticating users while finding application on addressing the process and transmission of messages. However, there are other algorithmic schemes that enhance the performance of user authentication and distribution of messages in the network. We start by presenting the authentication related messages.

In [148] the researchers propose the EAAP (Efficient Anonymous Authentication) algorithmic scheme, which targets on providing effective anonymous authentication for vehicles and RSUs existing in a VANET network. It makes good use of the bilinear-pairing cryptographic logic and detects between authentication processes for RSUs and vehicles (nodes). The TA (stands for Trusted Authority) enables parameters for signature and certificate generation in order to obtain secure authentication. There is the need for the various vehicles to register with the TA and get a DID, which stands for “dummy” identifier and can match it to their real ID. When a vehicle gets into the network, it produces an anonymous certificate, then it signs it and provides it with the appropriate parameters. On the other side, the RSUs receive their identifiers from the TA, register in the system and produce anonymous certificates via the use of a temporary secret key. The described mechanism is made to provide security against attacks towards authentication and supports location privacy, with the opportunity to detect and prevent attackers from the network.

In [149] the researchers made a dual authentication scheme called PPDAS (Privacy-Preserving Dual Authentication and Key Agreement Scheme), which realizes encryption schemes based on the IDs of the vehicles and public key encryption. The scheme brings the concept of reputation in vehicles, where each vehicle is connected with the TA (Trusted Authority) via the RSU communication. The vehicles produce pseudonyms that can be decrypted only through RSU and

TA by using bilinear pairing mechanisms. In order to prevent or avoid replay attacks there is use of time stamps. The MAC (Message Authentication Code) is realized so that there are no message tampering attacks. So, the authentication procedure is made from two levels. As a starting point, vehicles transmit their pseudonyms, public keys and time stamps to the Trusted Authority through the RSU. Then the TA confirms the time stamp and compares the computed MAC code with the accepted code. When there is success, the 1st part of validation is finished. The vehicles' rumor is then proved, based on specialized confidence matrices and a calculated average confidence level of the communicating doublet. The second authentication level is connected to the trust level of the vehicles. After the termination of communication, vehicles assess each other and inform the trust tables by using TA via the RSU.

In [150] the researchers make use of the co-operative authentication scheme which includes the communication between vehicles and RSUs for safe message transmission. The mechanism makes good use of binary key trees for key control and authentication. Vehicles periodically confirm accepted messages, authenticate themselves, and send their results with close vehicles, which then attest their results. This procedure enables for the verification of the messages' legitimacy within the network, whereas the RSU is in charge of retracting dangerous or non-valid users. The latter can occur by revealing group key only to users that are valid and inform them about retracted vehicles through the Revocation List. The current mechanism enhances the ability of repealing and sharing public keys in a vehicle network, moreover, to detect malicious users. A disadvantage of these mechanisms is the grown network traffic and time delay which is an effect of the communication among the vehicles, the RSU, and a Trusted Authority in specific circumstances.

In [151] the researchers propose a Trust Mechanism in order to Protect Privacy via the use of both Blockchain and Multi-Party assessment, widely known as TMPP-BMPE. A broadcasting of data is proposed based on Paillier encryption algorithms and also ECDSA (Elliptic Curve Digital Signature Algorithm) so that it can provide data protection. They realize both homographic encryption and ECDSA so that can offer data privacy protection during the transmission period. They propose a trust logic based on multi-party assessment. The trustiness of Cooperative Partners (CP) is totally assessed concerning estimation indexes gathered from many entities, thus minimizing the dangers of interacting with not safe CPs. Finally, a scheme of blockchanin-assisted trust management is constructed so that they can hurdle malicious tampering with trusted data. As the authors state, the TMPP-BMPE works well enough in the field of protecting data privacy and evaluates the reliability of CPs by preventing at the same time, data tampering. Furthermore, what they propose, offers valuable insights for security and trust in IoV environments.

In [152] the authors propose a light scheme for preserving privacy authentication which mitigates Trusted Authorities (TA) reliance, aiming at achieving privacy protection and authentication between the many nodes (vehicles) and TA in VANETs fields by improving the "forgetful" exchange algorithm. The analysis of the security makes crystal clear that the scheme they propose enhances privacy and can withstand attacks. Furthermore, the performance simulation and evaluation demonstrate that their idea uses low computational resources and low communication overhead, while at the same time ensures low authentication delay and loss in packet rate. The last indicates a much better total performance against other schemes that are proposed the time of writing the current PhD thesis (2024).

7.2.2 Secure message propagation

In [153] the researchers proposed a different architecture for obtaining safe messages in vehicular Cloud networks. The proposed architecture aiming safe messaging in vehicular cloud networks targets to improve authentication and message transmission performance in areas with increased density. The mechanism realizes symmetric and asymmetric encryption for key and message transmission. If a vehicle is not in the coverage of a RSU, its message request is sent to the RSU via close vehicles using the AODV (Ad hoc On-Demand Distance Vector) algorithmic scheme. When a vehicle is registered in a peripheral cloud, it gets a key pair and the public key of the RSU. Thus, via digital signatures the messages undergo authentication, and are encrypted using non-symmetric schemes, so that the RSU is able to accept and authenticate private information from the various nodes (vehicles). The RSU can team multiple messages and transmit them in big packets to the local Cloud service in order not to face increased network traffic and the problems it brings. Pseudonyms are used in order to realize anonymity of each vehicle. The latter changes, when the number of vehicles drops, which is an extension of Mix-Zones. The mechanism provides security against many kinds of attacks but does not take under consideration the calculation load on the RSU when the density exceeds certain values.

In 2021 the authors of this research [154] proposed an algorithm known as ALI (Anonymous Lightweight Inter-vehicle), which targets on providing message authentication as well as encryption in VANETs via the used of ECC. It realizes the ECQV, better known as Elliptic Curve Qu Vanstone algorithm, which uses the rationale of Elliptic Curves to transmit the vehicles' keys. In order to exchange symmetric keys, the algorithm exploits ECIES (Elliptic Curve Integrated Encryption Scheme), that makes use of Diffie-Hellman scheme and what is known as Hash-based Key Derivation (HKDF). The TA (Trusted Authority) sends the keys to nodes (vehicles) and RSUs during the registration phase via the ECQV algorithm and then encrypts the messages with ECIES for quicklier communication. Keys can be used and when they expire, users ask for new keys from the TA. RSUs via the use of encrypted identities enable the communication between vehicles and TA. RSUs send symmetric key to nodes (vehicles) in order to enable encryption. In order to achieve that securely, the symmetric key is produced using the public keys of the TAs. The algorithm needs a daily renew between vehicles and the TA in order to get this public key and guarantee key update. It also exploits ECC and offers capable authentication, identity privacy and authentication. But message authentication takes place by nodes, something that can cause delays as a result of their constrained computing power.

7.2.3 Secure algorithms that deliver the computational load on the network

In the following research [155] the researchers made effort in order to categorize the vehicle authentication operation via the use of Cluster Heads. The nodes (vehicles) are teamed based on criteria as location, speed and computing power and what is more known as Cluster Head (CH) is chosen inside each team. The TA (Trusted Authority) authenticates the CH and gives to it a GID (Group ID). Since there is the authentication on the CH, then it has the responsibility for the authentication of the rest of the nodes inside the area. The vehicles that belong to the cluster gather the GID from the CH that they have given them and authenticate themselves by exploiting it. After that, the vehicles exchange a symmetric session Key in order to enable more communication. Both the authentication and the key exchange use ECC via the ElGamal algorithm. This type of authentication supports confidentiality, authentication and privacy on ID, and at the

same time they are durable in MTIM (man-in-the-middle) attacks. It also mitigates the times of computation. It also enhances both the size and the number of demanded messages.

Researchers in the following work [156] proposed an approach which combines group authentication with secure message propagation in vehicular cloud networks. The algorithm makes good use of RSUs in order to verify message authentication and integrity before spread. Moreover, it can "discuss" leadership roles assigned to nodes (vehicles), by mitigating computational load on the RSUs. Vehicles which enter the region of the RSUs get a certificate, validate it and settle communication with the related RSU. RSUs can choose a GL (Group Leader) in order to achieve authentication and message grouping in order to reach efficiency, same to the ones presented by the research [155]. Group Leaders often send proof of leadership as well as their public key and work like RSUs in their area of operation. Nodes authenticate and exchange symmetric keys with the GL or RSUs. This occurs by using a number of Join or Accept messages which are encrypted via the use of asymmetric cryptographic algorithms. RSUs guarantee messages and are responsible for data propagation. The TA keeps a revocation list and keys renewal, while pseudonyms are also exploited in order to guarantee location privacy as well as anonymity. The algorithm improves security and efficiency in VANETs via the use of many cryptographic methods in order to block attacks.

In [157] the authors propose a mechanism for data sharing in private in VANETs through the use of Federated Learning (FL), using also local differential privacy. As the 1st stage, the various nodes (vehicles) realize local differential privacy techniques in their data before transmit them to the RSU. As the 2nd stage, there is a necessity to train the parameters of the model at the RSU and update the various trained weights with the server in charge for training. In order to assess the performance of the system, the authors evaluate it based on the precision and time needed for simulation including local and global parameters that were shared. Furthermore, they measure the performance of every client via the calculation of accuracy measurements that exist in each iteration. The outcomes of the experiments indicate that their framework manages the following:

i) Guarantees security against inference, attacking inference and gradient leakage.
ii) Presents advanced efficiency in comparison to other competitive solutions.

In [158] the authors propose the ALRS (Anonymous and Linkable Ring Signcryption) scheme, which represents for anonymous and linkable ring signcryption scheme for Location-Based Services (LBS) in VANETs. It provides privacy in data to the service providers and evaluates privacy in the vehicles of the users. Moreover, it contains linkability, something better known as the number of times that the same user needs a query. The latter occurs without the need to expose the ID of the vehicle user. Realization and detailed outputs show that the proposed ALRS scheme worked better than did other algorithms concerning communication and computation costs by keeping all the privacy requirements.

7.2.4 Blockchain Mechanisms in IoV Privacy Maintenance

In [159] the researchers propose a mix of Blockchain and Crowd so that it can provide Location Privacy Protection (BCS-LPP) in IoV fields. Starting with, they introduce blockchain into BCS-LPP aiming at preventing the leakage of users' data from third-party service focus. Secondly, the scheme offers workers with location privacy level options combined with Geolash encoding. In order to manage confidentiality in the privacy information of the location related to the workers, it supports order-preserving encryption. Finally, there is guarantee of the equity of workers'

involvement in various works, completed through the verification of sensing locations provided by the workers. The authors make use of actual data and make a comparison between the BCS-LPP and the existing mechanisms via simulations. As they state, BCS-LPP can provide the quality of data coming from the sensors, offer safety in terms of workers' location privacy data and enhance the fairness of the various users' engagement in the tasks.

In [160] the authors propose a scheme for offering privacy in vehicular networks through the use of blockchain. More especially, they produced an anonymous and auditable idea for data sharing through the use of Zero-Knowledge Proof algorithm (ZKP) so that they could guarantee the identity privacy of the vehicles. In the same sense, they maintain the data auditability in nodes for Trusted Authorities (TAs). They made an efficient multi-sharding mechanism that can limit costs of blockchain communication without the need to penetrate its security. The authors implemented a prototype of the idea and laboured many experiments and tests in simulators with it. Through the tests they found that their mechanism "reinforces" system security and data privacy, furthermore it mitigates communication time complexity $O\left(\frac{n\sqrt{m}}{m^2}\right)$ in contrast to other proposed sharding algorithms.

In [161] the researchers propose an idea, which is called dual blockchain based decentralized architecture for authentication of the nodes, moreover for secure and efficient communication inside the IoV network. As they state the usage of individual blockchains in order to authenticate and share messages makes the network to become more efficient and faster, while at the same time helps to reach the security of the network. Both blockchains make use of different approximations, thus using more security. The security and performance evaluation of the idea indicates that it can handle attacks with reduced computation cost, extended throughput, diminished delays in transmission and an extended vehicle verification rate in contrast to other existing solutions.

7.3 Analysis of the cryptographic protocols used

In the current section the four cryptographic algorithms employed, are analysed. More specifically we analyse the basics of advanced encryption standard (AES), elliptic curve cryptography (ECC), hyperelliptic curve cryptography genus 2 (HECC-2) and hyperelliptic curve cryptography genus 3 (HECC-3). The current analysis is based on discrete logarithm problem (DLP) that ECC, HECC-2 and HECC-3 make use of, in order to establish secure communication between 2 parties, making it impossible for a 3rd party to penetrate the secure channel and eavesdrop the messages exchanged. We give an initial idea of the mathematics used.

7.3.1 Advanced Encryption Standard (AES) Cryptographic protocol

The AES algorithm was initially proposed by Joan Daemen and Vincent Rijmen. The algorithm was chosen as the mainstream encryption scheme by the widely known NIST (National Institute of Standards and Technology) in 2001 [162]. There are many characteristics that make AES algorithm distinguishable from similar symmetric cryptographic schemes that are presented below:

- The AES algorithm is what is called a block cipher making use of the same key either there is need for encryption or there is need for decryption.
- The unencrypted text block size is 128 bits, equal to 16 bytes

- The size of the key can be 128/192/256 bits
- AES contains iterations, better known as rounds, linked to the key sizes. These are the following: (1) 10 rounds for a 128-bit key size (2) 12 rounds for 192-bit key size and (3) 14 rounds for 256-bit key size.
- AES includes 4 levels, which are the following: (a) SubByte, (b) ShiftRow, (c) MixColumn, (d) AddRoundKey. These stages exist with a continuous rationale and are implemented as rounds on a 4 x 4-byte state array. The plaintext (the text to be encrypted) is inputted in those rounds.

Below, there is a more careful view of each level:

(a) The **SubByte** stage demonstrates byte substitution including non-linear "discipline", which is implemented on every byte of the state array through an independent rationale.
(b) The **ShiftRow** stage is made of 4 rows. The starting row is not shifted, whereas the rest of the three rows are moved in a circular logic over 1 to 3 bytes.
(c) The **MixColumns** level uses the column-by-column logic in order to operate the state. Every column is used as a polynomial made of 4 terms in GF (28). Then, it goes through multiplication with a fixed a(x) modulo x^4+1 polynomial.
(d) The **AddRoundKey** stage operates the levels via the use of one of the sub-keys by implementing XOR (eXclusive OR) operation. The latter is realized between every byte of the subkey and each byte of the state.

7.3.2 Elliptic Curve Cryptography

The two famous mathematicians Koblitz and Miller, proposed in 1985, separately a group of points sourcing from an elliptic curve that was set on a finite field so that it could exploit the widely known Discrete Logarithm Problem in Cryptography. That happens, because the Elliptic Curve points cannot use the existence of prime factors which the Number Field Sieve takes advantage. The most optimized method that solves the Discrete Logarithm in the Group is known to have square root complexity, which makes the group appropriate for cryptographic schemes [163].

For using the Curves in order to build the desirable group, we must, firstly set the elements of the Group and the procedure that will be further used so that we can define the Group. The Curve described here and the points that set the Group are provided by the following equation:

$$E: y^2 = x^3 + Ax + B, \text{ where: } (A, B \in \mathbb{K})$$

Moreover, the curve must be non-singular, something that means that $-(4A^3 + 27B^3)$ must not exist in the Finite Field $\mathbb{K}$, to have a characteristic varying from 2 and 3. Thus, the elements of the group would consist of points which are set by the curve. The neutral element points to infinity.

Here, the process of Group is defined. From what can somebody observe from the curve's form, each element $P = (x_0, y_0)$, includes a reverse point, which is the second point that line $x = x_0$ intersects the curve, thus $P' = (x_0, -y_0')$.

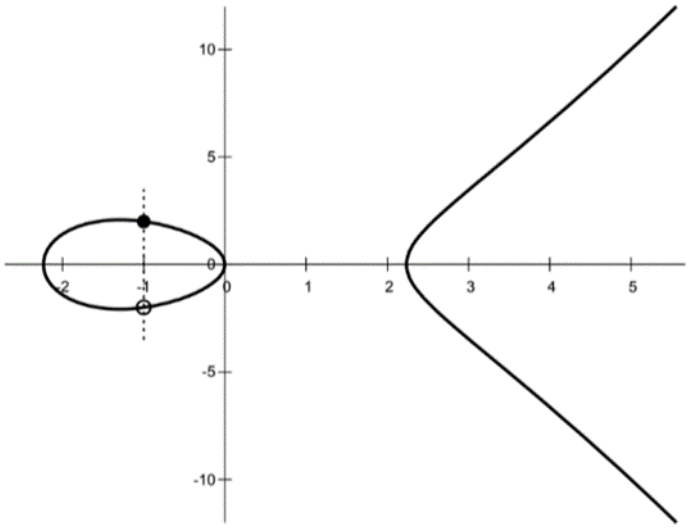


Figure 126 Reverse points of the Elliptic Curve [163].

In order to set the computation of Group, there are used 2 points on the curve. Let these points be *P* and *Q*. The line which is passing from the 2 points, intersects in a 3rd point the curve, let it be the *R*. Thus, we can set the process $P \oplus Q$ which gives the *R'* point, that represents the reverse of *R*, because *P, Q* and *R* are collinearly (**Figure 126**).

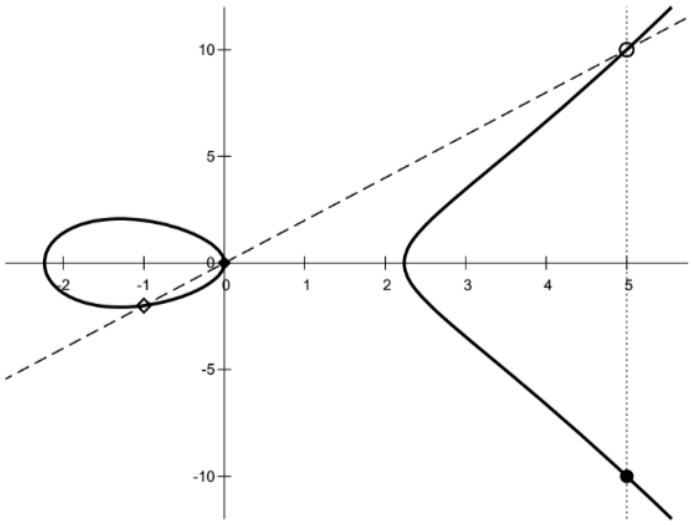


Figure 127 Rule of the Group for the Elliptic Curve points [163].

The following scalar multiplication is set so that we can use the Discrete Logarithm:

$$P \longmapsto [n]P = P \oplus P \oplus P \ldots \oplus P \text{ for n times}$$

The existence of Discrete Logarithm Problem is obvious. The symbol *P* stands for the generator element and *n* represents the exponent of the generator so that we could build the Cyclic Group with the deliberate characteristics of the uniform distribution (**Figure 127**).

Thus, in the Elliptic Curve cryptographic (ECC) systems, the element *n* which makes the scalar multiplication of the generator *P* can stand for the secret (private) key of the user and the result of the computation can represent the public key. So, in order for a hacker/attacker/intruder to gain the private key by analysing the public key, the only case is to solve the Discrete Logarithm Problem in the Cyclic Group that was aforementioned, something seriously difficult. On the other hand, the procedure of scalar multiplication is quite easy, because of the many mathematics techniques which are calculated with efficiency. It is more than important to choose the generator and the parameters of the curve very carefully, in order for the Cyclic Group to contain a prime number order, which is a little bit difficult. But, in ECC, there exist efficient techniques in order to make the parameters safe for the curves and the generator. The order of Cyclic Group is about $p + O(\sqrt{p})$, where *p* stands for the prime number that was selected for defining the Finite Field, in order to have an estimation of each curve's security. More specially, due to the fact that the best-known algorithms are able to solve the Discrete Logarithm have $O(\sqrt{p})$ complexity, where *p* has size of *n* bits, then in order to have 128-bit level security, the *p* should have *n/2* equal to 128, meaning 256 bits.

The Elliptic Curve Cryptography (ECC) is better known today for being used in cryptocurrencies and guarantee of security in various systems that contain constrained power and memory, as happens in embedded systems and in IoT, and also in areas related to digital signatures and digital certificates. Advanced research has been made in the current scheme and there exists an official data base which maintains secure improvements in the wanted calculations, with the result to be used easily in applications where there is no need to know how the scheme operates from a mathematical point of view. ECC can support small key sizes security, in relation to the large key sizes of the RSA.

The very careful selection of the needed parameters can guide to security gaps in the mechanism that will be selected for an application, if the developer who have chosen this, has no in-depth knowledge of the mathematics behind this protocol, and makes wrong estimations to that estimation. Furthermore, the DLP (Discrete Logarithm Problem) cannot defense Quantum Computers' attack. The consequence of using ECC is that it will face security problems in the future, because of the Quantum Computing attacks.

7.3.3 Hyperelliptic Curve Cryptography for genus ≥ 2

After 5 years of the Elliptic Curve Cryptography proposition, Koblitz proposed that Jacobian Curves could be used so that we could produce an appropriate Group. Elliptic Curves consist a sub-set of the Hyperelliptic Curves. To give an idea, Elliptic Curves are Hyperelliptic Curves for genus = 1, and in general when genus = 2, call them Hyperelliptic Curves.

In order to tail after the corresponding procedure of the defining of a cyclic Group which is based on Hyperelliptic Curve with $g \geq 2$, where *g*, stands for genus, it does not need to take it as process of the group, the procedure that was referred aforementioned, because this time every line intersects the curve in $2 \cdot g + 1$ points [163].

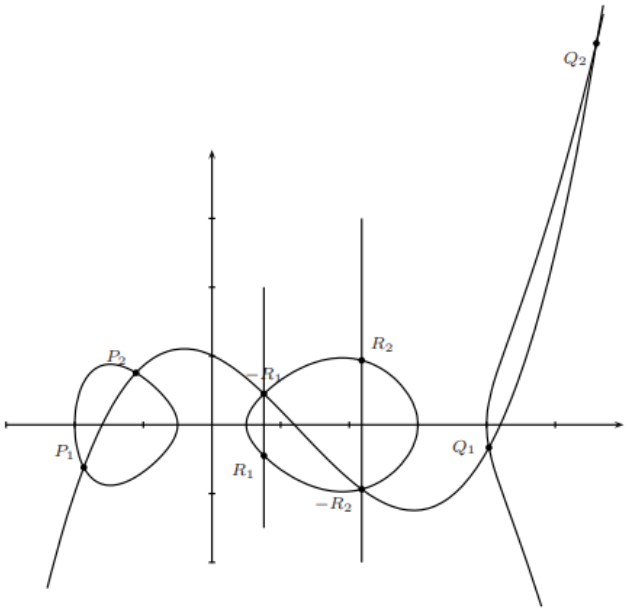


Figure 128 This is an example a g = 2 Hyperelliptic Curve, with $y^2 = f(x)$[163].

With the below equation we can denote a Hyperelliptic Curve genus g:

$$C: y^2 + h(x)y = f(x), h, f \, \epsilon \, K[x], \deg(f) = 2g + 1, \deg(h) \leq g, f \; singular$$

As it was described in the previous paragraphs, the curve must be what is called in mathematics non-singular, which is secured with the condition that none of the points is zeroing the partial derivatives. So, in order to format a Cyclic Group, the scheme that was proposed is to adjust a set of points where the summation tails after a function. For instance, when there is a curve like this in **Figure 128**, which someone can observe that it depicts a curve of genus 2, the points depicted and more specifically $R_1 = (x_{R1}, -y_{R1})$ and $-R_1 = (x_{R1}, y_{R1})$, in the curve $x = x_{R1}$, we can make the assumption that $R_1 \oplus (-R_1) = 0$. Continuing with the rest points $P_1, P_2, Q_1, Q_2, -R_1, -R_2$ result in a cubic function, so the end output is zero. So, as an element we can assume the elements that are the outcome of the sum of two elements (in general 2 points) and the computation between 2 points (*g* in general) elements gives as an outcome the result of the two points where there exists a cubic function $y = s(x)$, or in general *g + 1* order, which intersects the Hyperelliptic Curve.

The aforementioned class is named Order of the Divisor Class Pic_c^o of the Curve. So, in order to set the Class with the correct way, we contain again the point to infinity, that in reality is supposed to intersect in infinity each parallel line with y-axis. The class of the Divisor accepts many shapes, but, the one responsible for setting the Class and for producing the arithmetic is the Mumford representation. Thus, the components of the Class result in two polynomials $u(x)$ and $v(x)$ which obey in the following three characteristics:

- the u has to be singular
- $\deg(v) < \deg(u) \leq g \; (genus)$
- the u must divide accurately the polynomial $v^2 + vh - f$

As a result, the polynomial $u(x)$ can be described by the following equation:

$$u(x) = \prod_{i=1}^{r} (x - x_i)$$

where the x_i denotes the coordinate of the points that are selected in order to form the element. More specially for the P_i points that are engaged, known also as the support points, there has to apply that $P_i \neq P_\infty$, $P_i \neq -P_j$, for each $i \neq j$. The latter form is implemented in the Class of Reduced Devisors, only when $r \leq g$. The interesting part is the fact that we set the Class that shapes the Cyclic Group. Moreover, it is very considerable to set effective algorithms for the computation for the aforementioned Class. Many algorithms were made and optimized for the execution of this computation, but, as it can be observed, it is not easy to calculate in relation to the one which was set for Elliptic Curves. What makes the Hyperelliptic Curves suitable for cryptographic protocols, and Cryptography is that when we need to aim at a specific security level, the Finite field and the equivalent needed arithmetic, both of them must be in smaller numbers.

The latter occurs because the components of the Class consist of the g points but not from one point. So, the Class holds p^g size, in cases when p is quite large. So, guided by the calculation of the security level that was analysed in the Elliptic Curves part, a Hyperelliptic Curve with genus 2 can hit a 128-bit security level with a Finite Field of size 128-bit, while Hyperelliptic Curve g = 3 can hit similar security level with 86-bit. As it is obvious, while the genus enlarges, the Finite Fields size becomes smaller, as a result the keys that are going to be used, they get smaller and reach the same security level. The drawback is the existence of Index Calculus attack, which may be impossible to be implemented in Elliptic Curves, but it is able to solve the DLP in a meaningful time, in cases where the genus becomes larger. This is why g = 4 is not used in that type of cryptographic mechanisms [164].

Because HECC is not as dominant as happens with ECC, there are no many databases that they could set the various parameters that Curves need so that they could be selected. This is why, HECC is not used widely in Industry, because it is still an area of research. The algorithms suitable for computing the Class level, which is the outcome of the Reduced Divisor class, are not so effective something that is too difficult for the computation of HECC security.

One of the positive characteristics of small key size is that it makes it suitable for systems which contain limited resources, as it occurs with embedded devices and the widely known VANETs (Vehicular Ad-hoc Networks). The latter is one of the reasons of the intense research the current years around HECC.

7.4 Which was the followed solution

As it was presented in the related section about state of the art, the security scheme that was proposed by the researchers in [165] introduced the Group Leader cluster. The latter was used in the below approach so that it could simulate secure message scenarios on VANETs, something that was extended it order to realize ECC and HECC cryptographic schemes and various techniques for evaluating performance. The needed tools and software are analyzed, whereas the implementation of the network and the cryptographic algorithms are explained in the next paragraph.

7.4.1 Software used

Some widely known software packages were used in order to build the simulation environment for the estimation of the cryptographic schemes on VANETs. Network Simulator 3 (ns-3)[63], which is an open-source package, was used in order to simulate a sample VANET, moreover to manage the network protocols on the lower OSI levels. ns-3 supports a realization of the WAVE scheme stack, better known as WIFI 802.11p IEEE standard and the IEEE 1609. Moreover, it supports the routing algorithm which targets VANET and AODV use in order to evaluate the energy consumption on various network modules running in the simulation. In addition, in order to generate a more realistic traffic in the simulation, SUMO package was used so that there is simulation and analysis of road traffic. SUMO tool can cooperate efficiently with ns-3, via the use of XML files, which can be added to the ns-3 tool and generate realistic VANET scenarios. PyViz was used in order to have a visualization of the simulation. Again ns-3 tool cooperate with PyViz in order to provide a visualization of the VANET model.

7.4.2 Realization of the cryptographic algorithms

The section discusses and analyzes the design of the proposed approach, with an in-depth presentation of the cryptographic protocols used. Scheme [165] realizes many cryptographic schemes in order to ensure security in VANET communication nodes (vehicles). In the simulation that took place, we do not give emphasis on exchange of pseudonyms and key update or key revocation. We emphasized on the procedure of selecting and updating the Group Leader in the network and finally, the process of a node sending information to the RSU or GL about important events in the region. For this reason, symmetric algorithmic scheme is realized via the use of AES cryptographic scheme, whereas asymmetric encryption is realized via the use of ECC, HECC genus 2 and HECC genus 3. The latter was helped using also ElGamal scheme[64]. ElGamal demands the messages to be a cryptographic Group element, and more specifically a reduced divisor. For this reason, a solution of mapping (matching) text to a reduced divisor is also important to be realized. Moreover, ECDSA signatures and HEC ELGamal signatures take place. HECQV and ECQV (Hyper-Elliptic Qu Vanstone) certificates are in charge for key generation and distribution. Since there is no open-source library for Hyperelliptic Curves Cryptography specialized for genus 3 in C++ programming language, we built one for the current research.

As discussed above, the parts of the scheme [165] are based only on symmetric encryption, such as AES, in order to use the methods supported by C++ library, Crypto++. AES chosen to realize in the CBC mode, 16 bytes bits key and 128 bits block sizes. An initialization vector takes place and is sent with the symmetric key. To correctly send the keys and IVs to the network as strings, we used the following schemes: HexEncoder and HexDecoder libraries. Text which is encrypted is always a multiple of 16 bytes, because PKCS padding is used to manage input blocks.

Crypto++ supports more an API for executing ECC cryptographic schemes, such as scalar multiplication, point addition and realized techniques for signing and verifying messages via the

[63] https://www.nsnam.org/about/what-is-ns-3/

[64] https://en.wikipedia.org/wiki/ElGamal_encryption

use of ECDSA (Elliptic Curve Digital Signature Algorithm) signatures. Curve secp256r[65] and its parameters, such as base element or generator element, is supported by Crypto++, which is selected for the current simulation due to the fact that it generates 256-bit keys, so 128-bit security level. In order to realize ElGamal encryption/decryption, Scalar Multiply was used, for instance: Add and Subtract techniques of the used Crypto++ in order to generate the cipher text as a tuple *(a, b)*, where:

$$a = k \cdot G$$

$$b = k \cdot P + M$$

where:
k: stands for a random integer of Group order (256 bits)
G: represents the generator element
P: represents the public key
M: is the encoded text via the use of Elliptic Curves

The plain text is received by the following formula:

$$M = b - x \cdot a$$

where:
x: is the private key

In order to encode (not encrypt!) text as an Elliptic Curve point, the Koblitz scheme is realized [166]. Modular arithmetic is implemented via the use of NTL library[66]. Having got the plain text as an integer *x*, which is less than the Group Order, the computed Elliptic Curve Point: (x_1, y_1) is:

$$x_1 = x \cdot k + i$$

where:
$k = 1000$ and
$1 \leq i \leq 999$

The x_1 is computed by the previous formula until there is a quadratic residue[67], for instance:

$$y^2 = x_1 \, mod \, n$$

can give a solution. ECDSA signatures are produced via the use of Signer class of Crypto++ and verification is implemented via the Verifier class of Crypto++. The points are compressed in order to send only the *x* coordinate with one more byte so that there is a construction of *y* coordinate accordingly.

[65] https://neuromancer.sk/std/secg/secp256r1
[66] https://libntl.org
[67] https://en.wikipedia.org/wiki/Quadratic_residue

Key-pair production, distribution and verification are up to the ECQV certificate algorithm [167] for ECC, HECC genus 2 and HECC genus 3. The algorithm is fitted to the needs for HECC. ECQV contains 5 steps:

- ECQV_Setup
- Cert_Request
- Cert_Generate
- Cert_PK_Extraction
- Cert_Reception

In ECQV_Setup, the simulated nodes, which responses as the CA (Certificate Authority) produces a key-pair d_{CA}, Q_{CA}. The public key Q_{CA} should be known in each vehicle that takes part in the RSU. The Cert_Request step is when a node asks for a valid certificate from the CA. The latter produces a key-pair k_U, R_U and forwards the public key R_U to the CA accompanied with its ID. The Cert_Generate step, is when the CA receives the R_U from the vehicle (node), and the former generates a new key-pair $k, k \cdot G$. After that it calculates the

$$P_U = R_U + k \cdot G$$

and produces the certificate that includes P_U. CA also computes the

$$r = e \cdot k + d_{CA} (mod\ n)$$

where:
$e = Hash_n(Cert)$
n = Group Order

As the last part of the current step, CA sends the certificate and the parameter *r* to the node. When the Cert_PK_Extraction steps takes place, any vehicle that desires to extract the public key from the certificate computes the following equation:

$$Q_U = e \cdot P_U + Q_{CA}$$

The vehicle that asked for the certificate, during Cert_Reception part, it calculates its public as was described before. Moreover, it calculates its private key:

$$d_U = r + e \cdot k_U\ (mod\ n)$$

where:
$e = Hash_n(Cert)$

It then confirms if the following equation has effect:

$$Q_U = d_U \cdot G$$

As far as ECC is concerned, these operations take place via the Crypto++ API for ECC calculations, with scalar addition and multiplication. In the Hashing part the SHA3-256 class of Crypto++ library is exploited. In addition, when having to do about modular arithmetic the related class, meaning the ModularArithmetic class is used. This class is initialized on *n*, the Group Order.

The famous libg2hec[68] library was used in order to realize cryptographic operations of HECC genus 2. This C++ library is optimized for providing methods for producing and implementing operations on reduced divisors of HECC genus 2, capable for cryptographic computations. The techniques followed in the following sections were collected from [168]. The library is based on NTL C++ library optimized for mathematics' number theory. ELGamal and HECQV certificates were exploited for encryption and decryption operations. ECC used the previous same operations, but in this case the Group Elements were reduced divisors in Mumford scheme. ElGamal signature part was used for signing and verifying messages. There are same computations as ECDSA, but the techniques were realized from "ground". The generated signature is the tuple:

$$(a, b)$$

where:

$$a = k \cdot g$$

and

$$b = \frac{m - x \cdot f(a)}{k} (mod\ N)$$

where:
m: is the reduced divisor text (message)
x: stands for the private key
k: is a random number in the range [1, N)
N: is the Group Order
g: is the reduced divisor generator element
f: stands for the bijection of a diminished divisor in Mumford representation matched to an integer [1, N)

The following bijection was used:

$$f(u) = (u_1^2 + u_2^2)(mod N)$$

where:
u_1, u_2 represent the parameters of the u reduced divisor polynomial in Mumford analysis.

The "difficult part" in HECC is to select secure Hyperelliptic Curve parameters for the cryptographic operations, because a database for curves and parameters does not exist, as it occurs in the ECC domain. Also, there are no generalized methods for encoding and decoding, which are needed for ElGamal encryption and signatures, as far as the Jacobian of the curves are concerned. For this reason, only algorithms for specialized Hyper Elliptic Curves have been built. Concerning ElGamal signatures the Group Order must be known and must be prime or "near" prime. The cofactor must be known also. For this reason, the signatures of concerning HECC genus 2 a curve was selected from source [169]. To be more analytic the following should exist:

$$y^2 = f(x)\ in\ F_p\ with\ p = 5 \cdot 1024 + 8503491$$

with

[68] https://github.com/syncom/libg2hec/tree/master

$$f(x) = x^5 + 268281082283935564490073 6x^3 + 226591355295993102902116x^2 + 254767471595292971789991 8x + 479730995970848967305935 0$$

is selected. This generates the following Group of Order:

$$N = 249999999999941304386009994022094639661975160756 9$$

where:
N: is a prime number of 128-bits size.

The last equation, basically means that the curve's security level is at 128-bit level and can be contrasted with secp256-r1 curve that is selected for ECC computations. In order to handle the issue of encoding text to the Jacobian of the curve, a method proposed by researchers in [170]. More specifically the idea was to encoding integers as points to a specific family of Hyper Elliptic Curve of random genus. The authors, provide code in Sage so that there can be an execution of encoding techniques, that was modifies to C++ code via the use of NTL library. There is need to use a different curve, and to be produced by their algorithm for a known Finite Field of a specific p. But $p = 3 mod\ 4$ and $p = 7\ mod\ 8$. A solution is to use the following characteristic of the Field

$$p = 340282366920938463463374607431768211223$$

which represents a 128-bit number and can generate a Group of Order close to 256-bits. Next, the curve is generated based on the following algorithm [170] and finds implementation to all cryptographic operations, apart from signatures made by ElGamal. But, the specific Order of the curve is not known. Based on [170], text is linked to a point in Hyperelliptic Curve and after that 2 points are connected and a diminished divisor is generated. More specifically, the u and v polynomials are computed using the following equations:

$$u(x) = (x - x_1) \cdot (x - x_2)$$

$$v(x) = c \cdot x + d$$

where:

$$c = \frac{y_1 - y_2}{x_1 - x_2}$$

$$d = y_1 - c \cdot x_1$$

A more realistic scheme is to use divisor compression methods, thus the v polynomial can be built from the u polynomial in Mumford realization. Based on [168] and with the knowledge that $u \mid v^2 - f$, a compression method is used and the divisors are sent using only 256 bits for both u_1, u_2 parameters and one more byte is used for remaking information in v polynomial.

A new library for HECC genus 3 was built to satisfy the needs of the simulation. We were based totally on libg2hec[69] library, that was referred to a previous section plus the following research [168]. As someone can understand, the curve and divisor classes were changed in order to satisfy

[69] https://github.com/syncom/libg2hec/tree/master

the needed conditions for genus 3 curves and divisors. As far as the Group operations are concerned, the following techniques for scalar multiplication were exploited: SAM (Square and Multiply), NAF (Non-adjacent Form) and ML (Montgomery Ladder). They were used as were realized in libg2hec because they are genus "agnostic". Cantor's algorithm can be used without any changes for divisor addition and divisor doubling, because although the algorithm is really slow, it is genus "agnostic".

The following 2 algorithms were used for genus 3 divisor addition and divisor doubling [168]:

Algorithm 1: Cantor's algorithm
Input: Two divisor classes $\overline{D}_1 = [u_1, u_1]$ and $\overline{D}_2 = [u_2, u_2]$ on the curve $C: y^2 + h(x)y = f(x)$.
Output: The unique reduced divisor D such that $\overline{D} = \overline{D}_1 \oplus \overline{D}_2$.

1. $d_1 \leftarrow \gcd(u_1, u_2)$ $\qquad [d_1 = e_1 u_1 + e_2 u_2]$
2. $d \leftarrow \gcd(d_1, v_1 + v_2 + h)$ $\qquad [d = c_1 d_1 + c_2(v_1 + v_2 + h)]$
3. $s_1 \leftarrow c_1 e_1, s_2 \leftarrow c_1 e_2$ and $s_3 \leftarrow c_2$
4. $u \leftarrow \frac{u_1 u_2}{d^2}$ and $v \leftarrow \frac{s_1 u_1 u_2 + s_2 u_2 u_1 + s_3(v_1 v_2 + f)}{d} mod\ u$
5. repeat
6. $u' \leftarrow \frac{f - vh - v^2}{u}$ and $v' \leftarrow (-h - v) mod\ u'$
7. $u \leftarrow u'$and $v \leftarrow v'$
8. until $\deg u \leq g$
9. make u monic
10. return $[u, v]$

Algorithm 2: Addition on curves of genus 3 in the general case
Input: Two divisor classes $[u_1, v_1]$ and $[u_2, v_2]$ with $u_i = x^3 + u_{i2}x^2 + u_{i2}x + u_{i1}x = u_{i2}x^2 + v_{i1}x + u_{i0}$
Output: The divisor class $[u'', v''] = [u_1, v_1] \oplus [u_2, v_2]$ with $u'' = x^3 + u_2''x^2 + u_1''x + u_0'', v'' = v_2''x^2 + v_1''x + v_0''$

1. Compute resultant $r = Res(u_1, u_2)$ (Bezout)
$t_1 \leftarrow u_{12}u_{21}, t_2 \leftarrow u_{11}u_{22}, t_3 \leftarrow u_{11}u_{20}, t_4 \leftarrow u_{10}u_{21}$ and $t_5 \leftarrow u_{12}u_{20}$
$t_6 \leftarrow u_{10}u_{22}, t_7 \leftarrow (u_{20} - u_{10})^2, t_8 \leftarrow (u_{21} - u_{11})^2$and $t_9 \leftarrow (u_{22} - u_{12})(t_3 - t_4)$
$t_{10} \leftarrow (u_{22} - u_{12})(t_5 - t_6)$ and $t_{11} \leftarrow (u_{21} - u_{11})(u_{20} - u_{10})$
$r \leftarrow (u_{20} - u_{10} + t_1 - t_2)(t_7 - t_9) + (t_5 - t_6)(t_{10} - 2t_{11}) + t_s(t_3 - t_4)$
If $r \leftarrow 0$ perform **Algorithm 1**

2. Compute almost inverse $inv = \frac{r}{u_1} mod\ u_2$
$inv_2 \leftarrow (t_1 - t_2 - u_{10} + u_{20})(u_{22} - u_{12}) - t_s$ and $inv_1 \leftarrow inv_2 u_{22} - t_{10} + t_{11}$
$inv_0 \leftarrow inv_2 u_{21} - u_{22}(t_{10} - t_{11}) + t_9 - t_7$

3. Compute $s' = rs \equiv (v_2 - v_1)inv\ (mod\ u_2)$ (Karatsuba)
$t_{12} \leftarrow (inv_1 + inv_2)(v_{22} - v_{12} + v_{21} - v_{11})$ and $t_{13} \leftarrow (v_{21} - v_{11})inv_1$
$t_{14} \leftarrow (inv_0 + inv_2)(v_{22} - v_{12} + v_{20} - v_{10})$ and $t_{15} \leftarrow (v_{20} - v_{10})inv_0$
$t_{16} \leftarrow (inv_0 + inv_1)(v_{21} - v_{11} + v_{20} - v_{10})$ and $t_{17} \leftarrow (v_{22} - v_{12})inv_2$
$r_0' \leftarrow t_{15}, r_1' \leftarrow t_{16} - t_{13} - t_{15}$ and $r_2' \leftarrow t_{13} + t_{14} - t_{15} - t_{17}$

$r_3' \leftarrow t_{12} - t_{13} - t_{17}, r_4' \leftarrow t_{17}$ and $t_{18} \leftarrow u_{22}r_4' - r_3'$
$t_{15} \leftarrow u_{20}t_{18}, t_{16} \leftarrow u_{21}r_4'$ and $s_0' \leftarrow r_0' + t_{15}$
$s_1' \leftarrow r_1' - (u_{21} + u_{20})(r_4' - t_{18}) + t_{16} - t_{15}$
$s_2' \leftarrow r_2' - t_{16} + u_{22}t_{18}$
If $s_2' = 0$ realize **Algorithm 1**

4.Compute $s = \frac{s'}{r}$ and make *s* monic
$w_1 \leftarrow (rs_2')^{-1}, w_2 \leftarrow rw_1, w_3 \leftarrow w_1 s_2'^2, w_4 \leftarrow rw_2$ and $w_5 \leftarrow w_4^2$
$s_0 \leftarrow w_2 s_0'$ and $s_1 \leftarrow w_2 s_1'$

5. Compute $\mathbf{z} = \mathbf{su_1}$
$z_0 \leftarrow s_0 u_{10}, z_1 \leftarrow s_1 u_{10} + s_0 u_{11}$ and $z_2 \leftarrow s_0 u_{12} + s_1 u_{11} + u_{10}$
$z_3 \leftarrow s_1 u_{12} + s_0 + u_{11}$ and $z_4 \leftarrow u_{12} + s_1$

6.Compute $u' = \frac{\left[s(z+w_4(h+2v_1)) - w_5\left(\frac{u_{20} - v_1 h - v_1^2}{u_1}\right)\right]}{u_2}$
$u_3' \leftarrow z_4 + s_1 - u_{22}$ and $u_2' \leftarrow -u_{22}u_3' - u_{21} + z_3 + s_0 + w_4 h_3 + s_1 z_4$
$u_1' \leftarrow w_4(h_2 2v_{12} + s_1 h_3) + s_1 z_3 + s_0 z_4 + z_2 - w_5 - u_{22}u_2' - u_{21}u_3' - u_{20}$
$u_0' \leftarrow w_4(s_1 h_2 + h_1 + 2v_{11} + 2s_1 v_{12} + s_0 h_3) + s_1 z_2 + z_1 + s_0 z_3 + w_5(u_{12} - f_6) - u_{22}u_1'$
$- u_{21}u_1' - u_{20}u_3'$

7.Compute $v' = -(w_3 z + h + v_1) mod\ u'$
$t_1 \leftarrow u_3' - z_4$ and $v_0' \leftarrow -w_3(u_0' t_1 + z_0) - h_0 - v_0$
$v_1' \leftarrow -w_3(u_1' t_1 - u_0' + z_1) - h_1 - v_{11}$
$v_2' \leftarrow -w_3(u_2' t_1 - u_1' + z_2) - h_2 - v_{12}$
$v_3' \leftarrow -w_3(u_3' t_1 - u_2' + z_3) - h_3$

8.Reduce u', in example: $u'' = \frac{f - v'h - v'^2}{u'}$
$u_2'' \leftarrow f_6 - u_3' - u_3'^2 - u_3' h_3$
$u_1'' \leftarrow u_2'' - u_2'' u_3' + f_5 - 2v_2' v_3' - v_3' h_2 - v_2' h_3$
$u_0'' \leftarrow -u_1' - u_2'' u_2' - u_1'' u_3' + f_4 - 2v_1' v_3' - v_2'^2 - v_2' h_2 - v_3' h_1 - v_1' h_3$

9.Compute $v'' = -(v' + h) mod\ u_3$
$v_2'' \leftarrow -v_2'' + (v_3' + h_3)u_2'' - h_2$
$v_1'' \leftarrow -v_1'' + (v_3' + h_3)u_1'' - h_1$

$v_0'' \leftarrow -v_0' + (v_3'' + h_3)u_0'' - h_0$

10.Return $[u'', v'']$

Algorithm 3: Doubling the curves of genus 3 in the generalized method
Input: A divisor class $[u, v]$with $u = x^3 + u_2 x^2 + u_1 x + u_0$ and $v = v_2 x^2 + v_1 x + v_0$.
Output: The divisor class $[u'', v''] = [2][u, v]$

1.Compute resultant $r = Res(u, \tilde{h})$ where $\tilde{h} = h + 2v$ (Bezout)
$t_1 \leftarrow u_2 \tilde{h}_1, t_2 \leftarrow u_1 \tilde{h}_2, t_3 \leftarrow u_1 \tilde{h}_0, t_4 \leftarrow u_0 \tilde{h}_1, t_5 \leftarrow u_2 \tilde{h}_0$ and $t_6 \leftarrow u_0 \tilde{h}_2$

$t_7 \leftarrow (\tilde{h}_0 - h_3u_0)^2, t_8 \leftarrow (\tilde{h}_1 - h_3u_1)^2$and $t_9 \leftarrow (\tilde{h}_2 - h_3u_2)(t_3 - t_4)$
$t_{10} \leftarrow (\tilde{h}_2 - h_3u_2)(t_5 - t_6)$ and $t_{11} \leftarrow (\tilde{h}_1 - h_3u_1)(\tilde{h}_0 - h_3u_0)$
$r \leftarrow \big(\tilde{h}_0 - h_3u_0 + t_1 - t_2\big)(t_7 - t_9) + (t_5 - t_6)(t_{10} - 2t_{11}) + t_8(t_3 - t_4)$
If $r = 0$ use Cantor's Algorithm **Algorithm 1**

2.Compute almost inverse $\boldsymbol{inv = \frac{r}{h+2v} mod\ u}$
$inv_2 \leftarrow -\big(t_1 - t_2 - h_3u_0 + \tilde{h}_0\big)(h_2 - h_3u_2) + t_8$
$inv_1 \leftarrow inv_2u_2 + t_{10} - t_{11}$
$inv_0 \leftarrow inv_2u_1 + u_2(t_{10} - t_{11}) - t_9 + t_7$

3.Compute $\boldsymbol{z = \frac{f - hv - v^2}{u}\ mod\ u}$
$t_{12} \leftarrow u_2^2, z_3' \leftarrow f_6 - u_2, t_{13} \leftarrow z_3'u_1$ and $z_2' \leftarrow f_5 - h_3v_2 - u_1 - u_2f_6 + u_2^2$
$z_1' \leftarrow f_4 - h_2v_2 - h_3v_1 - t_{12} - u_0 - t_{13} - z_2'u_2$
$z_2 \leftarrow f_5 - h_3v_2 - 2u_1 + u_2(u_2 - 2z_3')$ and $z_1 \leftarrow z_1' - t_{13} + u_2u_1 - u_0$
$z_0 \leftarrow f_3 - h_2v_1 - h_1v_2 - 2v_2v_1 - h_3v_0 + u_0(u_2 - 2z_3') - z_2'u_1 - z_1'u_2$

4.Compute $\boldsymbol{s' = (z\ inv) mod\ u}$ (Karatsuba)
$t_{12} \leftarrow (inv_1 + inv_2)(z_1 + z_2)$ and $t_{13} \leftarrow z_1inv_1$
$t_{14} \leftarrow (inv_0 + inv_2)(z_0 + z_2)$ and $t_{15} \leftarrow z_0inv_0$
$t_{16} \leftarrow (inv_0 + inv_1)(z_0 + z_1)$ and $t_{17} \leftarrow z_2inv_2$
$z_0' \leftarrow t_{15}, r_1' \leftarrow t_{16} - t_{13} - t_{15}$ and $r_2' \leftarrow t_{13} + t_{14} - t_{15} - t_{17}$
$r_3' \leftarrow t_{12} - t_{13} - t_{17}, r_4' \leftarrow t_{17}$ and $t_{18} \leftarrow u_2r_4' - r_3'$
$t_{15} \leftarrow u_0t_{18}, t_{16} \leftarrow u_1r_4', s_0' \leftarrow r_0' + t_{15}$ and $s_1' \leftarrow r_1' - (u_1 + u_0)(r_4' - t_{18}) + t_{16} - t_{15}$
$s_2' \leftarrow r_2' - t_{16} + u_2t_{18}$
If $s_2' = 0$ use **Algorithm 1**

5.Compute $\boldsymbol{s = \frac{s'}{r}}$ and make *s* monic
$w_1 \leftarrow (rs_2')^{-1}, w_2 \leftarrow w_1r, w_3 \leftarrow w_1(s_2')^2$, and $w_4 \leftarrow w_2r$ note that $w_4 = \frac{r}{s_2'}$
$w_5 \leftarrow w_4^2, s_0 \leftarrow w_2s_0'$ and $s_1 \leftarrow w_2s_1'$

6. Compute $\boldsymbol{G = su}$
$g_0 \leftarrow s_0u_0, g_1 \leftarrow s_1u_0 + s_0u_1$ and $g_2 \leftarrow s_0u_2 + s_1u_1 + u_0$
$g_3 \leftarrow s_1u_2 + s_0 + u_1$ and $g_4 \leftarrow u_2 + s_1$

7.Compute $\boldsymbol{u' = u^{-2}[(G + w_4v)^2 + w_4hG + w_5(hv - |f)]}$
$u_3' \leftarrow 2s_1, u_2' \leftarrow s_1^2 + 2s_0 + w_4h_3$
$u_1' \leftarrow 2s_0s_1 + w_4(2v_2 + h_3s_1 + h_2 - h_3u_2) - w_5$
$u_0' \leftarrow w_4(2v_1 + h_1 + h_3s_0 - h_3u_1 + 2v_2s_1 + u_2(u_2h_3 - 2v_2 - h_2 - s_1h_3) + h_2s_1)$
$u_0' \leftarrow u_0' + w_5(-f_6 + 2u_2) + s_0^2$

8.Compute $\boldsymbol{v' = -(Gw_3 + h + v) mod\ u'}$
$t_1 \leftarrow u_3' - g_4$
$v_3' \leftarrow -(t_1u_3' - u_2' + g_3)w_3 - h_3$ and $v_2' \leftarrow -(t_1u_2' - u_1' + g_2)w_3 - h_2 - v_2$
$v_1' \leftarrow (t_1u_1' - u_0' + g_1)w_3 - h_1 - v_1$ and $v_0' \leftarrow -(t_1u_0' + g_0)w_3 - h_0 - v_0$

9.Reduce u', for instance: $u'' = \frac{(f - v'h - v'^2)}{u'}$

$u_2'' \leftarrow f_6 - u_3' - v_3' - v_3' h_3$

$u_1'' \leftarrow -u_2' - u_2'' u_3' + f_5 - 2v_2' v_3' - v_3' h_2 - v_2' h_3$

$u_0'' \leftarrow -u_1' - u_2'' u_2' - u_1'' u_3' + f_4 - 2v_1' v_3' - v_2'^2 - v_2' h_2 - v_3' h_2 - v_1' h_3$

10.Compute $v_2 = -(v' + h)$ mod u_2

$v_2'' \leftarrow -v_2' + (v_3' + h_3)u_2'' - h_2$

$v_1'' \leftarrow -v_1' + (v_3' + h_3)u_1'' - h_1$

$v_0'' \leftarrow -v_0' + (v_3' + h_3)u_0'' - h_0$

11.return

$[u'', v'']$

Concerning the cryptographic schemes, the similar techniques as occurred in HECC genus 2 were realized. What distinguishes them are the selected curves, the encoding and decoding as well as the bijection techniques, that were adjusted in genus 3 divisors. By excluding signatures, a Finite Field, such as the following prime was exploited:

$$p = 773712524553362671811952 23$$

One more time, the following equation exist:

$$p = 3 \; mod \; 4$$

and

$$p = 7 \; mod \; 8$$

where p equals 86-bit integer, that generates a Group Order of about 256-bits. The latter Field is used by the algorithm presented in [170]. So, that it can produce a curve capable of exploited by the cryptographic schemes. As far as the signatures are concerned, the following was selected [171]:

$$y^2 = x^7 + x^5 + 6218231719898953 \cdot x^3 + 8683773159487505$$

The Group Order is

$$N = 8 \cdot q$$

where:

q: stands for 168-bits prime number

The bijection f which is used for ElGamal signatures, in the current case is changed to:

$$f(u) = u_1^2 + u_2^2 + u_2^2 \; (mod \; N)$$

For the Jacobian encodings, based on [170], the latter is used in order to link text to HEC points. Furthermore, three points are teamed together in order to generate a valid diminished genus 3 divisor.

7.5 Realization of the proposed solution

The current section makes an in-depth analysis of the implementation of the cryptographic algorithms used in the experiments that took place, such as AES, ECC, HECC-2 and HECC-3, including the most important aspects by offering specifically selected C++ screenshots. Furthermore, details are provided regarding the implementation of the communication between the VANETs and the RSU, the simulation of the road traffic and for the energy usage of the realized scheme. All the code used throughout the various experiments including the linked metrics and the ns-3 files have been available for free for research reasons on the following github repository[70].

7.5.1 Cryptographic methods

The AES algorithm was exploited in all the test in the simulator so as to cover the need of symmetric encryption in the safe scheme[71]. It is linked with one of the other 3 cryptographic protocols depicted before, such as ECC, HECC-2, HECC-3. In order to provide support to the encryption, the generation of the keys and the IV vectors there was usage of the Crypto++ library. The Crypto++ is realized in C++ programming language and offers the below characteristics:

- Operation method: Cipher Block Chaining (CBC), for providing better security
- Key size: 16 bytes (128 bits) for offering speed in the operations
- 128-bits block size

In [172] the algorithms 1 and 3, the RSU or the GL have to generate a symmetric key and transmit it to the node which is about to join the region. With the key, the IV is also transmitted. The generation technique of the aforementioned is realized in C++ as demonstrated in **Figure 129**.

```
AutoSeededRandomPool prng;

SecByteBlock key(AES::DEFAULT_KEYLENGTH);
SecByteBlock iv(AES::BLOCKSIZE);

prng.GenerateBlock(key, key.size());
prng.GenerateBlock(iv, iv.size());

std::string keystr, ivstr;
HexEncoder encoder(new StringSink(keystr));
encoder.Put(key, key.size());
encoder.MessageEnd();

HexEncoder encoder2(new StringSink(ivstr));
encoder2.Put(iv, iv.size());
encoder2.MessageEnd();
```

Figure 129 Generation technique linked to AES algorithm

As a first stage, 2-byte blocks containing bytes are produced through the use of the random number generation library (AutoSeededRandomPool). Then the key and the IV vector are

[70] https://github.com/PanosDgs/Secure_VANET_HECC/tree/master

[71] https://en.wikipedia.org/wiki/Wi-Fi

remodelled into strings via the use of *HexEncoder* of Crypto++ library. The latter library remodels the byte blocks to strings.

After that the operation of encryption/decryption of an AES message is as demonstrated[72] in **Figure 130** and **Figure 131**.

```
    CBC_Mode<AES>::Encryption e;
    e.SetKeyWithIV(key, 16, iv);

    StreamTransformationFilter encfilter(e, nullptr,
BlockPaddingSchemeDef::PKCS_PADDING);
    encfilter.Put(in, size);
    encfilter.MessageEnd();
    encfilter.Get(out, size+16-size%16);
```

Figure 130 Encryption of an AES message.

The function is fed with 2 buffers made out of bytes, an out buffer and an input buffer, the buffer size, the key and the IV in string format. Since the key and the IV remodelling is implemented again in bytes, as a consequence the encryption method as CBC AES is given as input to the encryption in a *StreamTransfrmationFilter.* The output size can be even bigger in cases when the input size is not a multiple of 16 bytes, because it is used the *PKCS Padding* technique so that it can transform the input size as an integer multiple of the block size.

In the decryption period, the opposite approach is followed. The remodelling of the key and the IV in Bytes are the same, as it is crystal clear from **Figure 130**.

```
        CBC_Mode<AES>::Decryption d;
        d.SetKeyWithIV(key, 16, iv);

        StreamTransformationFilter decfilter(d, nullptr,
BlockPaddingSchemeDef::PKCS_PADDING);
        decfilter.Put(in, size);
        decfilter.MessageEnd();
        decfilter.Get(out, size);
```

Figure 131 Decryption of an AES message.

The parameters of *decrypt_message_AES* are similar as before: however, the output is the decrypted message this time. When an error occurs in the decryption because of wrong key or IV, the method *decfilter.Put* throws an Exception which is collected by a *try…catch* command and then is printed.

For the realization of ECC, the *Crypto++* library was used one more time, which offers as ready many cryptographic functions. But the ElGamal is not inside, so it was realized via the *Crypto++*'s API. Furthermore, the Koblitz technique and the message Encodings in the Curve were produced. Another realization was linked to the ECQV algorithm, because it was needed in the certificates, where also ECDSA was used. The curve which was selected is one of the most well-known, secure

[72] https://www.cryptopp.com/wiki/Advanced_Encryption_Standard

and fast curves, such as the famous secp256r1[73], which makes keys of 256 bits size, so 128 bits security level.

The realization of HECC-2 was based on this software[74] and distend this to make it capable for ns-3 and IoV fields. However, at the time of writing the current section (2024), no HECC-3 library existed. For this reason, via the use of the *libg2hec*[75] library as the main reference and by following the algorithms that were presented before in the book, classes, divisors and techniques of Curve genus 3 were implemented with C/C++ programming language. The related realization of HECC-3 for the various tests that have taken place on the simulator is available in the following github repository[76]. For instance, the *g3hcurve::update* method that was implemented for the validity check of the Curve is presented in **Figure 132** as seen below:

```
//Set is_genus_3
if( deg(fpoly) == 7 && deg(hpoly) <= 3)
        is_genus_3 = TRUE;
else
        is_genus_3 = FALSE;
```

Figure 132 Validation of the HECC genus 3 Curve.

The code snippet demonstrated in **Figure 132** targets on checking whether the polynomial *f* is up to order $(2 \cdot g + 1)$ and if the polynomial *h* is about order 3 (= *g*). The proof that the Curve must be non-singular stays the same as *libg2hec* via the use of NTL library. The random Curve production is accomplished with the same way as it happens with *libg2hec*, by just altering the order from 5 to 7 for *f* polynomial and from 2 to 4 for *h* polynomial of the Curve.

After that the divisor's curve genus 3 class is realized. The following code, depicted in **Figure 133** checks about valid divisor.

```
OK = OK && IsOne( LeadCoeff(upoly) ); // (1)

OK = OK && ( deg(upoly) <= genus ) && ( deg(vpoly) < deg(upoly) );
// (2)

OK = OK && IsZero(( vpoly*(vpoly + curve_g3.get_h())

                              - curve_g3.get_f() ) % upoly ); //(3)
```

Figure 133 Realization of divisor's Curve genus 3.

The most crucial part of the implementation is the divisor arithmetic. The techniques for scalar multiplication are irrelevant to the Curve's genus, that's why some rules have to be updated from *libg2hec* library. The techniques which are offered are the known as SAM, the NAF and the ML. The algorithms which have to be implemented are the "addition" and "doubling" divisors. By using addition through Cantor's algorithm, referring to algorithm 14 from the following work [173] is "genus-free" and can include the arithmetic in genus 3 curves. The disadvantage is that it is very

[73] https://neuromancer.sk/std/secg/secp256r1

[74] https://github.com/syncom/libg2hec

[75] https://github.com/syncom/libg2hec/tree/master

[76] https://github.com/PanosDgs/Secure_VANET_HECC/tree/master

slow, and for this reason the 14.52 and 14.53 of the same work [173] were used, which are the most optimized with less and more "light" calculations. The implementation, only shows the residual arithmetic calculations of the two previously discussed algorithms in C++ programming language.

7.5.2 Road traffic simulation

SUMO software was used in order to realize road traffic and transform an area existed in reality, to an XML file for network implementation and route planning. The SUMO tool produced random routes for the various nodes, generated 400-second traffic simulation and distributed a vehicle for each 0.5 seconds. The latter built an XML (Extensible Markup Language) file, that was then transformed into a .tcl format capable to be exploited by both ns-2 and ns-3 simulators via the SUMO trace exporter. The number of nodes (vehicles) in ns-3 simulation was fixed into 63 vehicles, plus the extra RSU nodes fixed manually. Specific start and exit times were adjusted for the vehicles' simulation when entering or exiting an entry point. Moreover, we setup WAVE (Wireless Access in Vehicular Environments), devices on each vehicle via the use of suitable ns-3 classes, using power modifications realized for each transmitted packet.

7.5.2 VANET communication realization

The following packages were used in order to realize the communication between the nodes and the RSUs: Crypto++, libg2hec, NTL and g3hec. Communication was managed via the packet layer of WAVE protocol, retrieving metrics with no more delay by operating on higher OSI (Open Systems Interconnection) layers, such as network and application. We setup callback functions in order to use packets coming to vehicles and handle them according to different status they enclosed, such as: Join, Accept, Extract_Symmetric, Receive GL Proof, Receive Vehicle Information concerning source in [165]. Responses were organized to be sent randomly in a range of 0 to 3 seconds, so that we could better exploit the total bandwidth of the mean and also mitigate as better it can the collisions. The status of all the nodes, keys, CA data and public key of the nodes who insert in area with the corresponding certificates in a specific area are stored by the RSU. The latter also keeps the total number of nodes which have joined, the GL identifier, the exchanged symmetric keys between the nodes and various data related to ECC and HECC that were used in the encryption part. As far as the vehicles are concerned, they store the following information:

1. Their keys
2. CA
3. RSU public keys
4. GL public keys
5. Certificates
6. Status of communication
7. Symmetric keys
8. IV vectors
9. Parameters related to curve

The communication begins with the RSU broadcasting its certificate with period equals to 2 seconds. All the nodes (vehicles) start in RECEIVE_CERT state. When the RSU certificate is gathered, vehicles verify it, accept the RSU public key, store the key and transmit back a Join packet to the RSU, altering to RECEIVE_ACCEPT_KEY state. RSU computes the response, changing the state of the node initially to RECEIVE_ACCEPT_KEY, then ON_SYMMETRIC_ENC after transmitting the ACCEPT answer. The density of the vehicles increases after 120 seconds, guiding

to congestion. RSU randomly chooses a GL (Group Leader), sends GL Proof of Leadership, the GL goes to GROUP_LEADER_INFORM state after that to IS_GROUP_LEADER. At this point it is very critical to note that ECC and HECC do not permit encryption of the Proof of Leadership message, and as a result the RSU offers to the GL its own certificate, signed by the PK of the RSU. GL conveys decrypted message to the network every 2 seconds. Old and new vehicles watch Join and Accept operation with the GL (via RECEIVE_ACCEPT_GL and ON_SYMM_GL states). As a final step, vehicles transmit an INFORM message randomly when they receive a symmetric key. GL gets Inform messages and transmits them to the RSU.

7.5.3 How the energy model was realized

NS-3 supports techniques for estimating the energy consumption on nodes in regard of the energy that a real network protocol, such a WAVE would consume when the node communicates with the network. More specifically the NS-3 WIFI Radio Energy Model of NS-3 was exploited in order to estimate the battery's capacity reduction after one or more of the following WIFI states: 1) Idle, 2) CcaBusy, 3) Tx, 4) Rx, 5) ChannelSwitch, 6) Sleep, 7) Off. Each WAVE network was initialized with 1000 Joules. Then each energy consumption was subtracted from the energy level. After every significant step of communication that was referred in previous paragraph, the energy that remains is subtracted from the initial energy level that was stored before the communication has taken place, thus the energy consumption is measured (calculated).

7.6 Assessment of the experiments

The current chapter delegates the results that came out from the various experiments that took place via the following three asymmetric cryptographic protocols that were analyzed in a previous section (7.4), ECC, HECC (genus 2), HECC (genus 3).

The simulation contained 63 vehicles and 1 RSU, covering the entire area. The various messages were sent with maximum available power and with the aim to decrease the collisions from concurrent transmission and was selected to implement a simple setback of the feedback up to 3 seconds. The time period was set to be large enough for tangible systems, but it was utilized in order to better supervise the communication. The measurements targeted cryptographic techniques. The latter technique rises the summing time, and as a result it increases the energy that period, because it influenced the operational time. So, in some communication parts, someone can see that the scheme uses more power than others.

For measuring the time, we used the well-known function: chrono::high_resolution_clock::now() placed at the start and at the end of each calculation. In the diagrams that follow, they depict the mean time of the duration and the energy. A byte buffer calculation took place in the measurement of the parameters before sending message to the node. Thus, the measurements did not contain the size of the header which is added by the WAVE protocol.

For the simulation the following components were used:

- Linux Ubuntu 20.04.6 with host Windows 10
- 8 GB RAM
- 4 cores (Intel Core i5-10300H, clocked @ 2.50 GHz)

- NS-3.30
- SUMO 1.16.0
- NTL 5.5

7.6.1 Assessment framework

Chosen well-known software equipment have been exploited in order to produce the right simulation environment for the assessment of the cryptographic methods on VANETs. The widely known open-source tool named Network Simulator 3 (NS-3)[77] has been taken advantage targeting on the simulations of sample VANET and also manage the related underlying network schemes. NS-3 offers a realization of the WAVE protocol, which is also known as WiFi 802.11p IEEE standard and the IEEE 1609 standard. It supports also the routing algorithm modeled for VANETs, AODV and solutions for calculating the energy consumption on network devices in the tested simulations. Moreover, in order to produce a realistic traffic model, there was usage of another open-source software tool, the SUMO. The latter is specifically used in order to simulate and analyze road traffic, while it can cooperate with NS-3 via the use of produced XML files, and input them to NS-3 for creating VANET with realistic scenarios. Last but not least, the PyViz was used, which is supported by NS-3 for offering a simple way for VANET visualization.

7.6.2 Duration of the various cryptographic operations

To start with, the times of the executions of the various cryptographic realizations were measured. More specifically the following parameters were assessed:

- Key-pair generation
- Certificate public key extraction
- Message decryption
- Message encryption
- Signature generation
- Signature verification
- Decoding
- Certificate private key reception
- Certificate generation
- Encoding

The outcomes of the aforementioned measurements include all the three cryptographic schemes that are referenced in this chapter, meaning: ECC, HECC-2 and HECC-3. All these results are depicted in **Figure 134**.

[77] https://www.nsnam.org/about/what-is-ns-3/

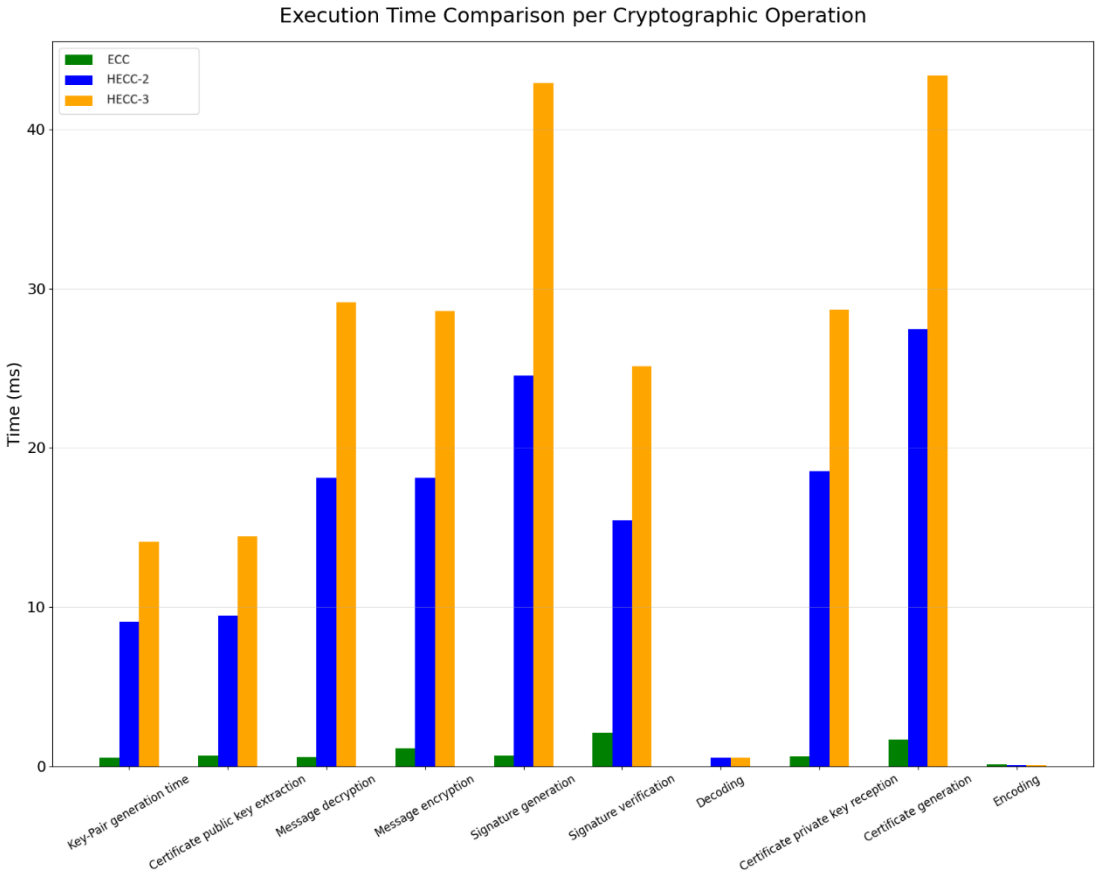


Figure 134 Times for various cryptographic realizations in ms.

In **Figure 134** as someone can observe, for the same 128-bit security level, ECC outperforms HECC g =2 and HECC g = 3, and more specifically ECC is 16 times faster than HECC g = 2 and 27 times faster than HECC g = 3.

As can someone see in **Figure 134** the outputs are similar in computing certificates and key extraction on ECQV mechanism in them. The production and the key extraction made through ECC outperforms HECC genus 2 by 15 times, and outperforms HECC genus 3 by 25 times. Key making is the slowest procedure, then follows the secret key extraction and the fastest among the three is the public key extraction, which is 3 times faster than the key production.

The outcome of the superiority of ECC in contrast to HECC happens because ECC algorithm is more mature and optimized by the researchers involved. HECC on the other hand, is quite modern idea without many optimizations. HECC contains very complex mathematics for a smooth transition of the algorithms into C++ code, and a lot of try and expertise have to be used in order to achieve satisfactory results in same the same sense of ECC.

Looking ECC, it is faster once again, against HECC g = 2 and HECC g = 3 concerning encryption and decryption. And to be more precise, ECC is nearly 13 times faster than HECC g = 2 and nearly 21

times than HECC g = 3. And of course, encryption is a bit faster than decryption. All these are depicted in **Figure 134**.

The ECC mechanism manage to overcome the other two implementations. But, the difference in the signature validation is restricted in relation to the other two cryptographic schemes. Thus, this happens due to the fact that there is a choice of different curves for the signatures guided by the HECC, that is part of 84-bit security level, while the ECC scheme gives back 128-bit security level. All these are depicted in **Figure 134**.

As far as the message encoding and decoding realizations durations are concerned, the encoding which is mainly uses Koblitz algorithm for ECC is quite slow in relation to HECC-2 and HECC-3 [173] as presented in **Figure 134** (encoding and decoding charts). To be more precise, when encoding takes place by using ECC seems to be around 38% slower than HECC-2 and HECC-3. Concerning the decoding duration, it seems to be around 50% faster in ECC in relation to HECC-2 and HECC-3.

7.6.3 What size consume the exchanged messages

As far as the communication that took place in communication, the following kinds of messages were realized:

- **RSU_CERT_BROADCAST:** This is the certificate which RSU sends with some period.
- **VEHICLE_SEND_JOIN_RSU:** The "Join" message which a node (vehicle) transmits to the RSU, when there is need to subscribe to the coverage area of the RSU.
- **RSU_ACCEPT:** The answer of the RSU to the message "Join" of the vehicle. It contains the cryptographic symmetric key.
- **RSU_INFORM_LEADER:** The RSU's message it transmits, so that it can update a chosen vehicle by the GL. It contains the Proof of Leadership.
- **GL_LEADERSHIP_PROOF:** The message broadcasted periodically from the GL, so that it can prove it is valid and attract nodes (vehicles) to "Join" him.
- **VEHICLE_SEND_JOIN_GL:** The "Join" message which the vehicle transmits to the GL, in order to subscribe to the related GL.
- **GL_ACCEPT:** This is the Group Leader's response to the vehicle, when the latter sends the "Join" message. It also contains the cryptographic symmetric key.
- **VEHICLE_INFORM:** The message which a node (vehicle) sends so that it can update its position. It updates the system evaluation.

As a result, the size of each message has been measured through the experiments. So, many experiments have taken place in which the sizes of the messages (that were presented above) have been assessed. The results of those experiments are presented in **Figure 135**. The results are presented for all the three cryptographic schemes, meaning ECC, HECC-2 and HECC-3.

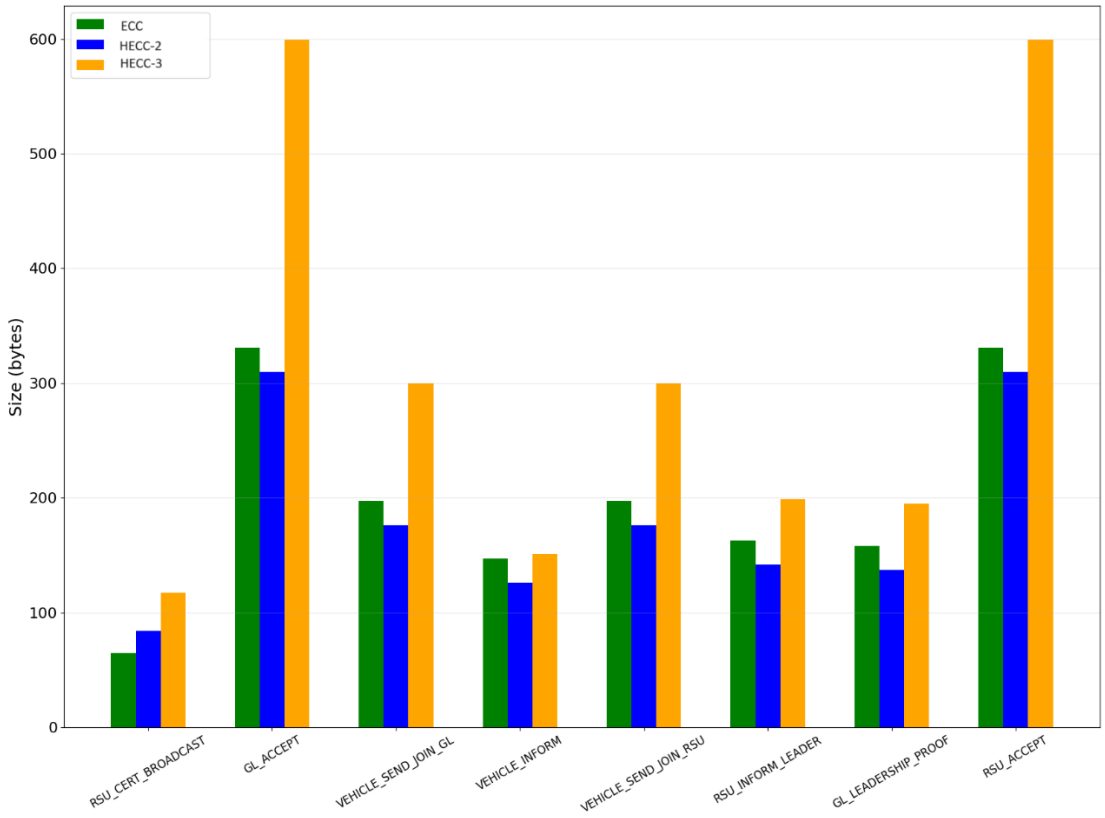


Figure 135 Size of exchanged messages when using the ns-3 simulation software. The sizes are depicted in Bytes.

As it is more than clear, the HECC-3 cryptographic algorithm outcomes the largest message GL_LEADERSHIP_PROOF in size, while the same message for ECC is quite smaller, with the HECC-2 been the smallest. Concerning the size of VEHICLE_SEND_JOIN_GL, the HECC-3 shows the largest size, while HECC-2 shows the smallest size for the same message. As far as the GL_ACCEPT message is concerned, someone can identify that the results are similar to HECC-3 for the largest size, with ECC being about the half of the related message size, while the HECC-2 outputs the smallest results of the three cryptographic schemes (**Figure 135**). Concerning VEHICLE_INFORM message size the rationale is the same for the ECC and HECC-3 algorithmic schemes (around 150 Bytes), whereas the HECC-2 outcomes a bit smaller VEHICLE_INFORM messages around 125 to 130 Bytes.

As shown in **Figure 135**, the messages sent and received when using ECC or HECC-2 consumes around the same size, while HECC-3 algorithmic scheme outputs messages of larger size. Someone can observe that in GL_ACCEPT and RSU_ACCEPT messages. Generally talking the messages' sizes are of quite the same size. In case there was use of similar level of HECC-2 signature, the related messages would have result of the similar size. If we need to accomplish the same size in genus 2, the rationale is to exploit the divisor compression. In HECC-3 it was a bit puzzle to make use of compression implementation, so that there is the transmission of information including data about the Curve in RSU_CERT_BROADCAST. The last action is more obvious, but it rises the size of the information. An interesting thing is that RSU_CERT_BROADCAST, GL_LEADERSHIP_PROOF and VEHICLE_INFORM messages were transmitted in some frequency, as occurs in non-simulated systems.

7.6.4 Energy consumption

The energy consumption was selected to be sensed in levels. Those levels were grouped in the same time durations. We measured the energy consumption in the side of the vehicles, but not in the infrastructure. The various levels are described below:

- **RECEIVE_CERT:** It refers to the process of reception and the computation of the RSU certificate (procedure related to reduced time-durations).
- **EXTRACT_GL_PROOF:** It refers to the process of reception and the computation of GL leader's proof (procedure related to reduced time-durations).
- **RECEIVE_ACCEPT_SYMMETRIC:** It is related to the procedure of transmission of messages in order to join and also the procedure of "Accept" reception from the side of RSU.
- **EXTRACT_JOIN_SEND_ACCEPT:** It is related to the procedure of "Join" message which GL receives from a node (vehicle) and also the transmission of the "Accept" feedback.
- **EXTRACT_INFO_GL:** It is related to the process of eliciting the message "Inform" which the GL receives from a message.

The diagrams in **Figure 136** and **Figure 137** depict the mean consumption of every level measured in Joules.

In **Figure 137**, someone can observe that the consumption of energy in RECEIVE_CERT message is no more than 200 μJ for ECC, HECC-2 and HECC-3. Concerning EXTRACT_GL_PROOF, the related power consumption is inside the range 300 μJ to 400 μJ. **Figure 137** shows the RECEIVE_ACCEPT_SYMMETRIC power consumption than is in the range of 1100 Joules to 1200 Joules for ECC, HECC-2 and HECC-3. The EXTRACT_ACCEPT_JOIN_GL_SEND_ACCEPT message power consumption falls inside the range of 600 mJoules to 900 mJoules. It is more than clear that EXTRACT_INFO_GL message power consumption is in the range of 300 mJoules to 700 mJoules.

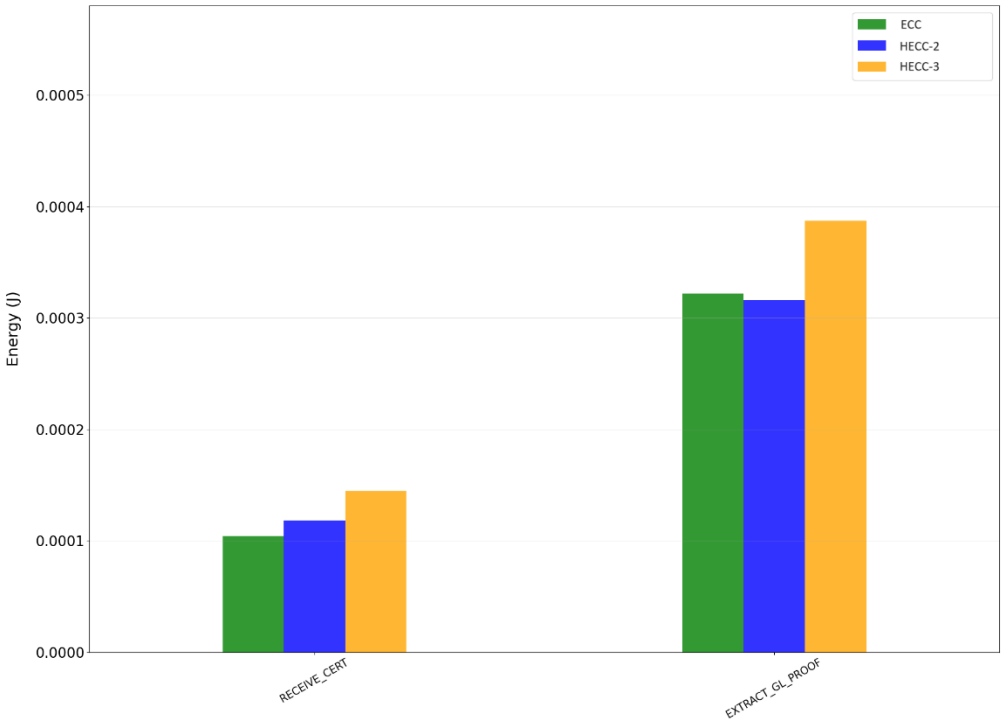


Figure 136 Consumption of energy in very small-time realization.

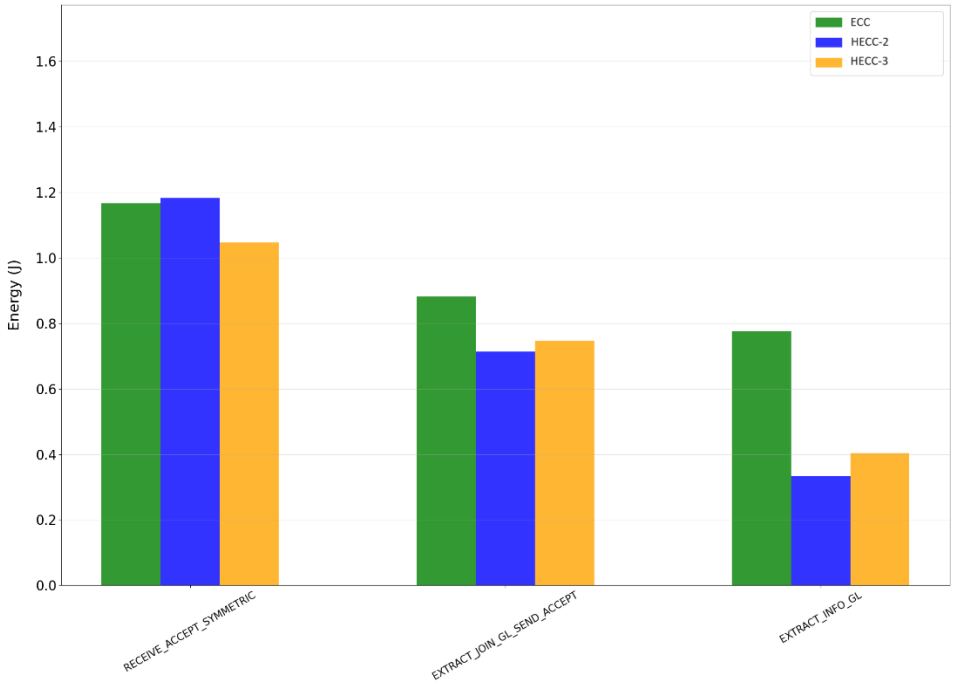


Figure 137 Energy consumption of the various operations, given in Joules.

As it is seen from **Figure 137**, HECC g = 3 consumes most of the energy for reduced time procedures, while the rest procedures, the ECC consumes the most energy among the other two cryptographic schemes. To give the general rationale (idea), the energy consumption is linked to message sizes. The latter occurs because the larger is, it needs fragmentation, so it produces more state switches in the device which transmits. Because the cryptographic algorithms that were selected from the messages they consume small size in the WAVE protocol packets, there is no need for fragmentation.

7.7 An innovative device for encrypting messages through Zigbee module via the use of AES and ECC

The current section demonstrates a realized device which transmits and receives messages of wireless network, and be more precise via the Zigbee module on ISM (Industrial Scientific Medical) band. The device uses 2 types of Arduino microcontrollers, the Arduino MEGA 2560 R3 and the Arduino GIGA R1. It also uses a TFT screen hat, connected to the Arduino MEGA 2560 R3, a voltage translator, Xbee Zigbee module(s), a USB splitter, DC power supply for each device (there are two devices, each one for every operator) and finally a USB keyboard.

7.7.1 The message encryption/decryption device

For the experiments, 2 different algorithms were used. The 1st one was then AES and the 2nd one was based on modified ECC algorithm, in order to be capable to encrypt and decrypt text messages. The following github repository[78] was selected for the AES implementation. Whereas for ECC the following library was used[79]. Both of the libraries support execution of code in Arduino CPUs.

An LCD with the below characteristics was connected to the Arduino MEGA 2560 R3:

- LCD Type: TFT
- LCD Interface: SPI
- LCD Controller: ILI9486
- Touch Screen Type: Resistive
- Touch Screen Controller: XPT2046
- Colors: RGB, 65K colors
- Resolution: 480x320 (Pixel)
- Aspect Ratio: 8:5
- I/O Voltage: 3.3 - 5V

The purpose of the LCD was to display the sent and the accepted messages of each user. The main idea is that every user chooses an encryption key, which is known to the other side, via a safe environment, and uses the key so that it can encrypt the messages transmitted to the other side or decrypt the received messages, so that they can read the decrypted text.

[78] https://github.com/DavyLandman/AESLib

[79] https://github.com/ShubhamAnnigeri/tinyECC-ArduinoIDE

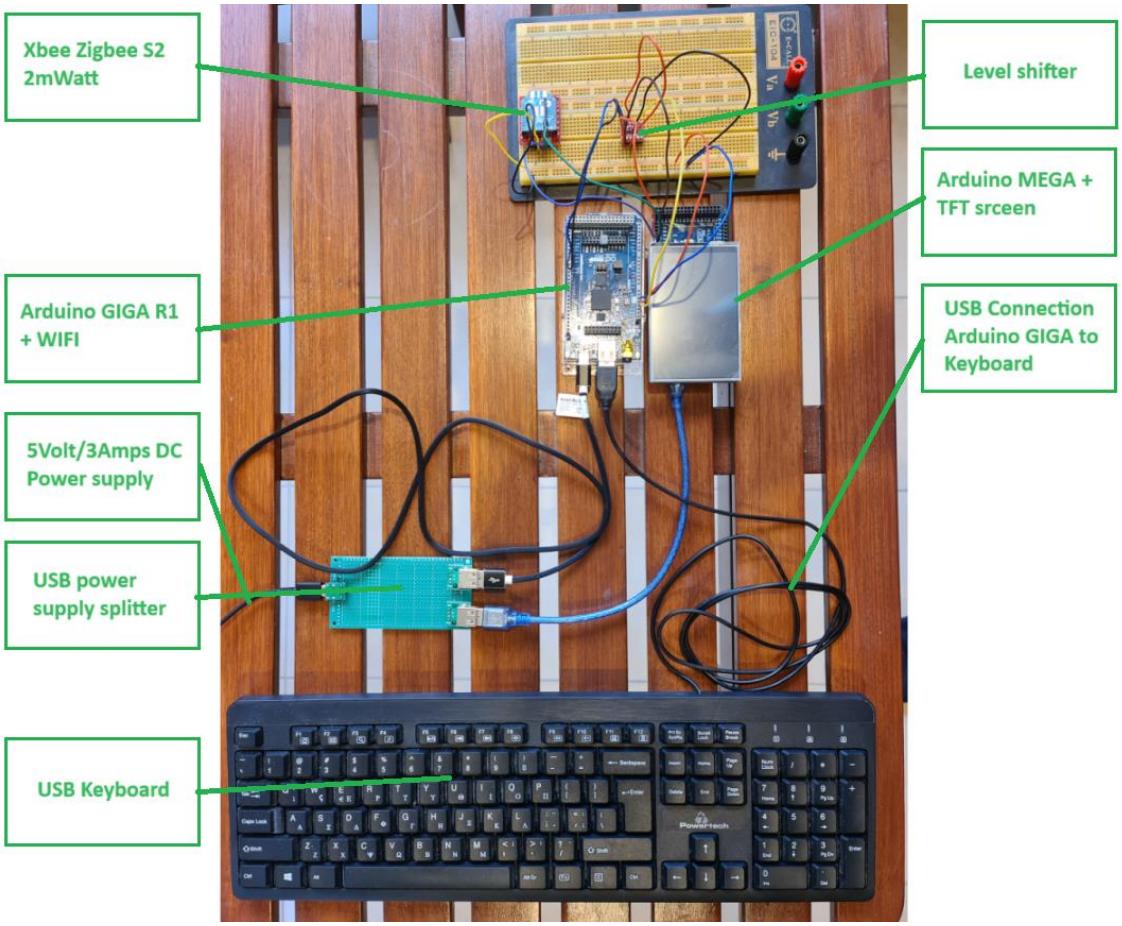


Figure 138 The encryption/decryption device with the related connections to the rest electronics.

The (new released) Arduino GIGA R1 was used because it incorporates support for USB interface, thus a USB keyboard could be connected. In the other side the Arduino MEGA 2560 R3 is attached with the TFT screen. As a result, the operator can observe the received decrypted messages. As it is widely known Arduino MEGA 2560 R3[80] is an open-source choice when someone needs to realize IoT experiments. It uses the ATmega2560 microcontroller and it incorporates UART protocol and various I/O pinouts, and a CPU clocked @ 16MHz.

Arduino GIGA R1[81] is a modern module suitable for IoT experiments with significantly more powerful processor than Arduino MEGA 2560 R3. It makes good use of a 32-bit dual core ARM Cortex CPU clocked at 480MHz and 240MHz, for the 1st and the 2nd core. It supports many I/O pins and many protocols, for instance 4xUARTs, 3xI2C, 2xSPI, CAN, 2Mbytes flash memory. It needs 3.3 Volts in order to operate both for power and logic. The latter explains the use of the SparkFun voltage translator[82], that makes reality the communication between the Arduino MEGA 2560 R3 and the Arduino GIGA R1. The chose of Arduino GIGA R1 was the only guaranteed solution so that there was use of USB keyboard.

[80] https://store.arduino.cc/products/arduino-mega-2560-rev3

[81] https://grobotronics.com/arduino-giga-r1-wifi.html?sl=en

[82] https://www.sparkfun.com/products/12009

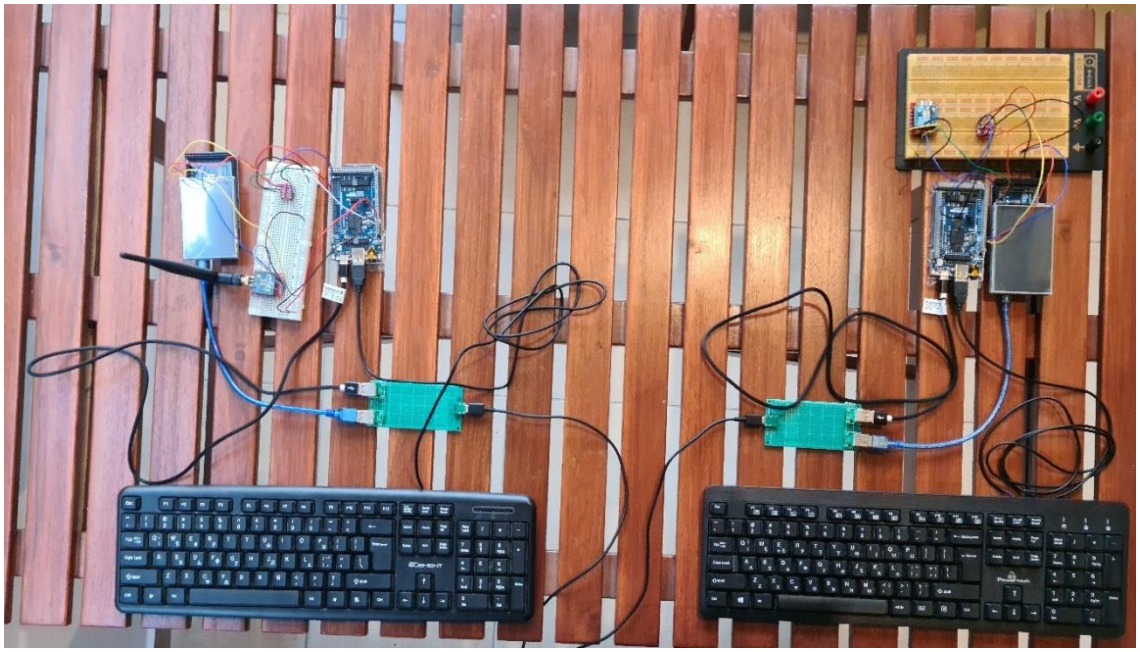


Figure 139 The experiment for 2 users. Each user can type the message which is sent encrypted to the other user wirelessly via the Zigbee wirelessly. The user that receives the message can read the decrypted message in their TFT screen.

There is use of a USB voltage divider in order to power supply the two Arduino modules, the Zigbee module and the SparkFun voltage translator. The Zigbee module can support up to 120 meters wireless coverage at LoS. Zigbee protocol supports the low-power rationale for data communication at 250kbps data rate. It needs 3.3 Volt to operate and consumes an average current at 40mA. This is another reason for using voltage translator, so that it can exchange information with Arduino MEGA 2560 R3, which needs 5 Volts to operate. The whole device needs 5 Volts DC power supply, moreover the supply can provide up to 3 Amperes, which of course is never reached, since the construction is low-power. The power supply is connected to a USB divider as depicted in the **Figure 138** and **Figure 139**.

Figure 140 shows the image of one of the two TFT screens when an encrypted message is received and the consequent decryption that takes place via the use of AES algorithm. The same logic is depicted in **Figure 141** where the message is received encrypted and then decrypted via modified ECC asymmetric algorithmic scheme.

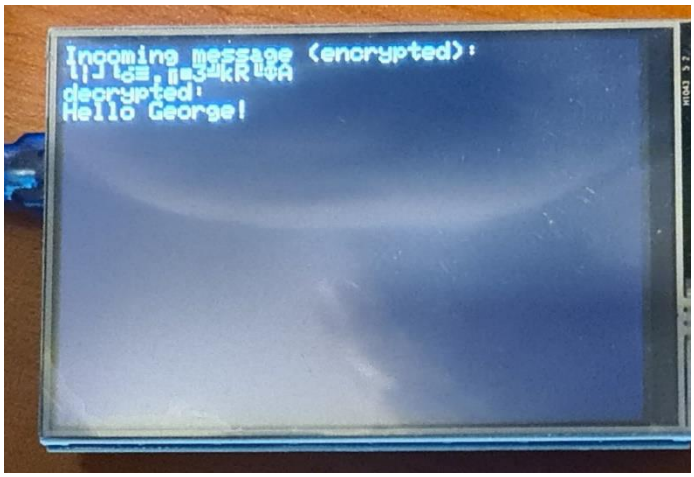


Figure 140 The interface of the device's TFT screen, when the user gets a decrypted message, when AES symmetric scheme is used.

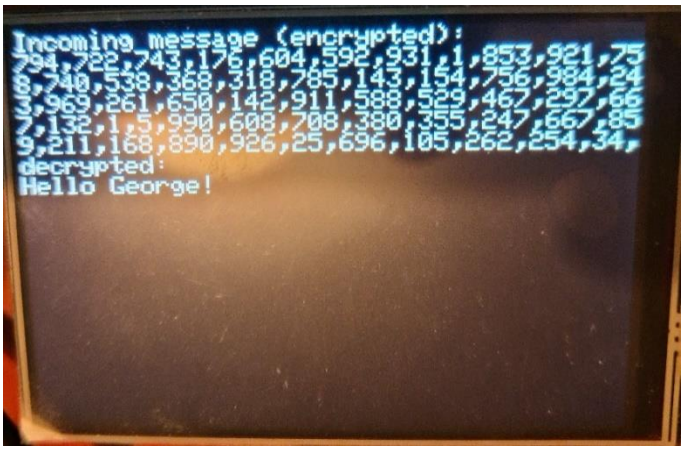


Figure 141 The interface of the device's TFT screen, when the user gets a decrypted message, when modified ECC asymmetric scheme is used.

Concerning the encryption and decryption times, some measurements took place, both for AES and ECC algorithms. As far as the AES is concerned, the time for encrypting the message: "Hello George!" took 1028 μsec while decrypting of the message took 1292 μsec. Concerning the modified ECC, encrypting the same message ("Hello George!") took 3861 μsec and decrypting the message took 1721 μsec. **Figure 142** depicts a comparison diagram between encryption or decryption of the same message, when using AES or modified-ECC executed on the Arduino MEGA 2560 R3 microcontroller.

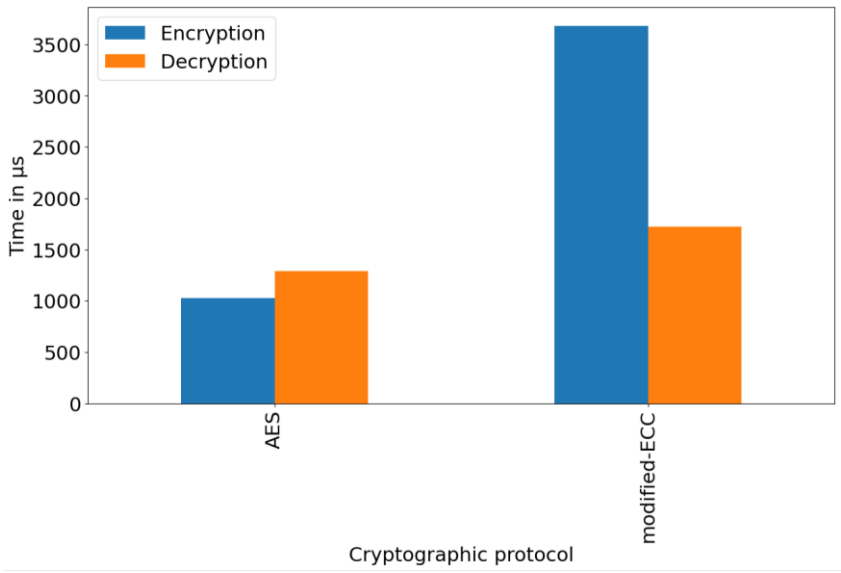


Figure 142 This figure demonstrates a comparison between the AES or modified-ECC when encrypting and decrypting the same message, when using Arduino MEGA 2560 R3 microcontroller.

An oscilloscope was used in order to gather the pulse-train of the unencrypted message (plaintext) (**Figure 143**) and the encrypted text (ciphertext) (**Figure 144**) that exits the Arduino MEGA 2560 R3 in order to input the Zigbee and then transmitted over the air to the receiver. As it is obvious from **Figure 143** the pulse is not more than 5 Volts. It is too difficult to depict the whole message due to the fact that the pulse-train is too large for the boundaries of the image.

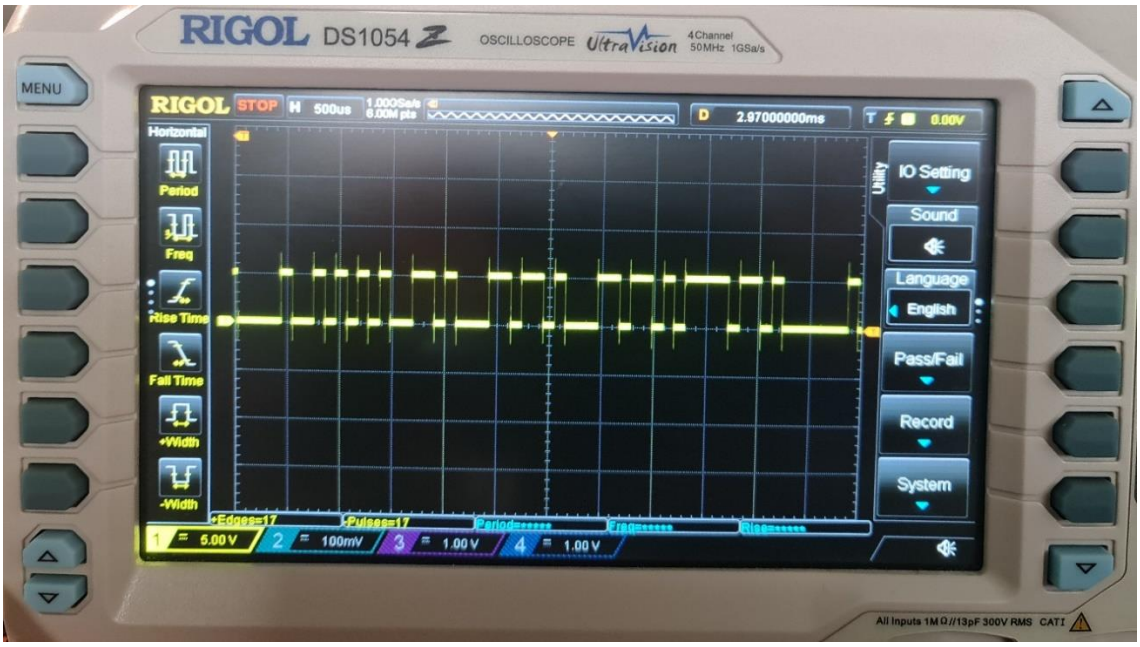


Figure 143 The electrical footprint of the plaintext while being sent from the USB keyboard to the Arduino MEGA 2560 R3 module. Then it undergoes encryption via either the use of AES or the modified-ECC algorithm.

The bitrate on which the Arduino modules communicate is fixed at 9600 bps. This occurs for Arduino MEGA 2560 R3, Arduino GIGA R1 and Zigbee module, that were analyzed in previous paragraphs.

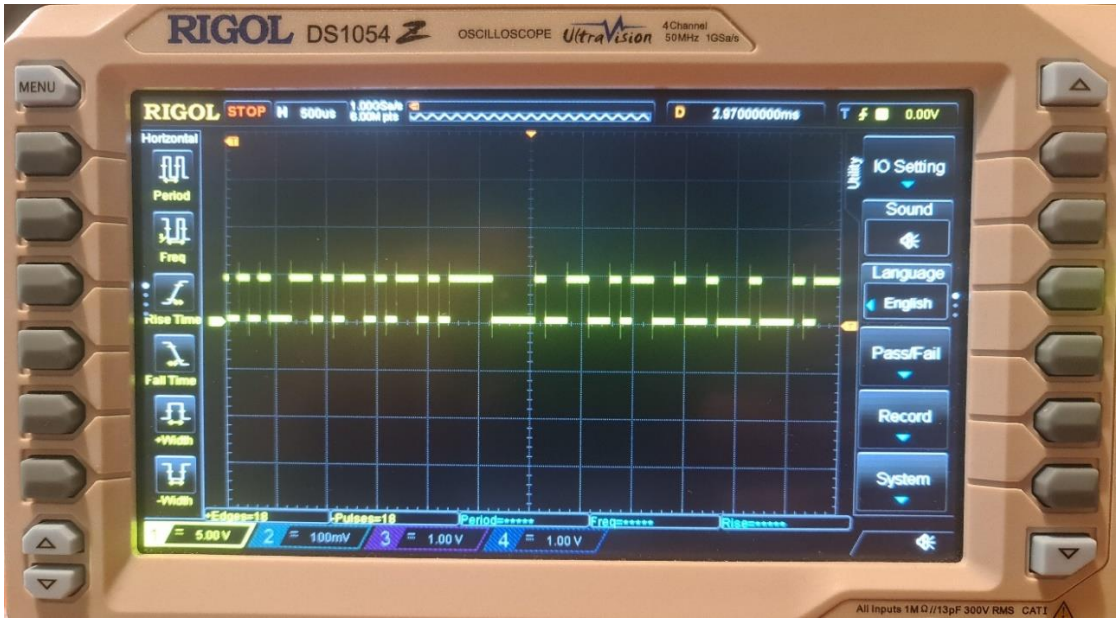


Figure 144 The footprint of the AES-encrypted message before be fed to the Zigbee module and then be transmitted wirelessly.

The following logic was followed. We made use of Arduino GIGA R1 so that we gather the metrics using either the AES or the modified-ECC protocols. Arduino GIGA R1 cannot be connected with the TFT screen due to incompatibility of the pins. The Arduino GIGA R1 was connected to the laptop in order to collect the metrics via the Serial reading of the Arduino IDE open-source software. The results of encryption and decryption for AES protocol are presented in **Figure 145**. The time durations for encryption and decryption of the modified-ECC protocol are also presented in **Figure 145**. From the results it is crystal clear that Arduino GIGA R1 supersedes Arduino MEGA 2560 R3. More precisely, for encryption using AES, Arduino GIGA R1 needs 34 μsec and Arduino MEGA 2560 R3 needs 1028 μsec. For decryption with AES, Arduino GIGA R1 needs 50 μsec and Arduino MEGA 2560 R3 needs 1292 μsec. As far as modified-ECC is concerned, Arduino GIGA R1 needs 3753 μsec for encryption, whereas Arduino MEGA 2560 R3 needs 3681 μsec. Concerning decryption with modified-ECC scheme, Arduino MEGA 2560 R3 needs 1721 μsec and Arduino GIGA R1 needs 2 μsec. It is crystal clear that Arduino GIGA R1 supersedes Arduino MEGA 2560 R3 in most metrics as far as the encryption/decryption processes are concerned related to AES and ECC cryptographic algorithms.

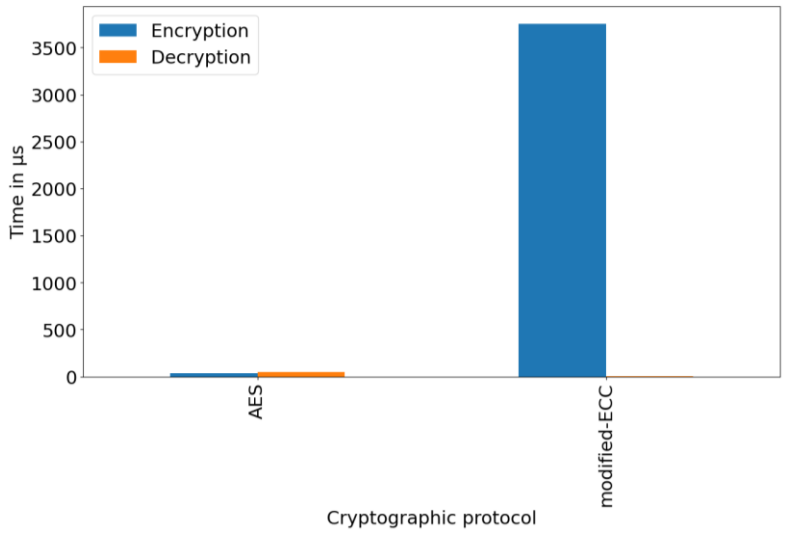


Figure 145 The time durations for encryption and decryption for the message: "Hello George!" realized in Arduino GIGA R1, using either AES scheme or modified-ECC.

7.8 Conclusions and future work

The current chapter made an introduction to the use of Hyperelliptic Curve Cryptography for improving privacy in IoV. The main idea is quite modern, although there are papers in literature that discuss about various techniques targeting on IoV privacy. However, HECC has not been used in such areas yet. Many experiments took place in ns-3 simulator where many entities, such as vehicles, RSUs and GLs were established. The main goal was to obtain the maximum privacy in response to many and different attack models. We studied the following cryptographic schemes, ECC, HECC-2 and HECC-3.

Many metrics have been taken place in order to assess the proposed idea. More specifically the following were tested: (1) key-pair generation time, (2) certificate public key generation, (3) message decryption, (4) message encryption, (5) signature generation, (6) signature verification, (7) decoding, (8) certificate private key reception, (9) certificate generation, (10) encoding. More metrics took place targeting the size of the exchanged messages and also the energy consumption for message exchange and message generation.

Furthermore, the current chapter delegates a modern approach for exchanging information through messages via Zigbee in the ISM free band. This occurs via the help of Arduino modules and suitably used electronics. The current proposition uses either AES symmetric cryptographic scheme or modified-ECC asymmetric cryptographic scheme. There are also metrics that depict the duration of encryption and decryption processes. The assessed metrics used also included the time duration of encryption/decryption regarding message exchange. The evaluation output shows that AES algorithm is faster than the modified ECC algorithm in the encryption and decryption tasks in most of the metrics gathered, for both Arduino modules used.

The results of the metrics show that in most of them ECC behave better than HECC-2 or HECC-3. But there are limited cases where HECC is better than ECC. This is because ECC has been greatly improved by the research community in recent years and performs better in most parameters. However, HECC is not so mature as occurs with ECC, moreover it has not undergone optimization to run fast, apart from the fact that it makes good use of smaller key sizes in relation to ECC for the same level of security. Furthermore, HECC contains the use of very complex mathematics, making it another serious reason why ECC finds more applications in real life. It is believed that if HECC is optimized as far as C/C++ programming is concerned, even with VHDL code, there would be a chance for better metrics results, much better than ECC.

As far as future extensions are concerned, there could be improvements in HECC performance. Initially, the arithmetic of HECC can be improved via a more modern library related to modular arithmetic. Another issue is that the NTL 5.5 can be a cause of delays and more specifically in scalar multiplication. Special curves could be used [174] which have shown in the past better performance in their arithmetic. As it is claimed in the following research work [175] the best selection of the parameters can improve, in a high degree, the performance of HECC. Another aspect, and more exactly in the area of embedded systems, there could also be used systems such as FPGAs or ASICs accelerators in real systems.

Moreover, the selection of Hyperelliptic Curves is constrained due to the existence of generalized algorithms for coding messages that aim at the curve. The scheme [176] that was used is considered to be the most generalized scheme that exists in the current literature and does not let the freedom to choose something else for the specific curve groups. One very important aspect is to identify which of the groups of curves can be cryptographically applied, so that they can generate a class with an order of what is called as “near” prime numbers. As a result, fragmentation algorithms of Jacobian points in hyperelliptic curves can be used for identifying the most secure. Also, the production of secure curves via the use of the CM method will output more choices of secure curves with known order. Thus, there will exist curves of the similar security level as well as in signatures for better comparison and study.

This page was intentionally left blank.

Part 3: Discussion and Future Plans

As it is obvious from the current PhD thesis, there are countless applications in the IoT that have solved significant challenges and are extremely useful in real life. Many experiments were realized, both in real case scenarios and computer simulations, allowing us to learn a lot about various less explored aspects and find interesting solutions to real challenges within IoT mechanisms.

We analysed the rationale and various details of IoT and Machine Learning in precision agriculture. A scheme was implemented in order to sense and evaluate different factors in a laboratory experiment. We sensed and logged temperature, UV radiance, soil moisture and air humidity. Through the use of a sophisticated Recurrent Neural Network - Long Short Term Memory (RNN-LSTM), we were able to forecast weather conditions, so the user could identify when there was need to irrigate the plant or farm field (in cases of scaling up). This way the user could save water resources and money by avoiding unnecessary/excess irrigation. Energy is a valuable/scarce resource in farms, therefore we also proceeded to an analysis of different IoT modules and the related wireless systems, in order to identify ways to optimize energy consumption. The Internet of Things has fully entered the agriculture sector, as there are numerous benefits for all involved actors, while there are plenty of affordable commercial sensors, as well as microcontrollers ranging from simple 8-bit 16 MHz CPUs (Arduino), to multicore ARM 64-bit CPUs clocked at 1.2 GHz (Raspberry Pi), available to everyone. Experiments have shown that multicore CUDA GPUs, and very sophisticated ASICs existing in Jetson Nano and Google Coral TPU respectively, can accomplish the same inference schedule by using less time and RAM memory in respect to a Raspberry Pi. Over the last few years, we have seen that IoT and Machine Learning models can cooperate in precision agriculture and provide interesting results. In the current thesis there is an extensive analysis of how a CNN model can be fed with satellite images and output predicted results concerning the soil salinity in a rice farm field. An RNN-LSTM network can also help in predicting environmental conditions and let the user decide if they should irrigate their farm, or postpone the irrigation due to fore coming hot days. We elaborated on the effects of Machine Learning models, and more precisely how the Convolutional Neural Network (CNN) model behaves when executed in different processing architectures. We used 3 SBCs that incorporated different processing units (CPU, GPU, TPU) in the inference part on image analysis related to leaves' diseases. Our research focused mainly on CPU-, RAM-, and swap memory usage, as well as temperature and energy consumption. Furthermore, we experimented extensively with Arduino IoT modules in rice and maize farms, in cooperation with Machine Learning and more specifically CNNs and RNN-LSTMs.

Another application we studied in agriculture is the auxiliary use of linear and multiple regression with ground IoT nodes in order to identify the average soil salinity in the whole area on a rice field. Linear and Multiple Regression were used for farm metrics' analysis, especially in rice fields farms. This application uses models output from the analysis of various Vegetation Indices and a specific model building. As a result, every new UAV image fed to the application and selection of the related VI or all the VIs, can give a soil salinity estimation of the field. We also performed extensive research in knowledge extraction working on the resin/rubber collection device. The pine resin has many applications in various fields in our everyday life. It is more than necessary for the farmer

to monitor pine resin. There is also a pioneer device analyzed towards resin and rubber collection presented based on Arduino microcontroller and various sensors, that transmits to the end user information about the environmental conditions of the resin/rubber collection via GSM/GPRS or via Xbee Zigbee. The device depicted in the current PhD thesis demonstrates a pioneer IoT device that can weigh the gathered resin from the pine tree, moreover it can inform the user about temperature and humidity as well as the existence of rain, in real-time via Zigbee protocol or near-real time via SMS messages.

We also performed experiments within the realm of the Internet of Vehicles (IoV), where security is a critical factor. More precisely, we simulated an IoV network of vehicles in an ns-3 simulator, where different asymmetric cryptographic protocols (NTRU, ECC, HECC-g2, HECC-g3, RSA) were analysed. The first idea was to use the MixGroup model with the various asymmetric cryptographic protocols, such as RSA, ECC, NTRU and the symmetric AES protocol, in order to provide IoV privacy and not let an attacker gather precious information. We observed metrics of encryption/decryption times, message sizes, signature generation times, signature verification times, exchange handshake sizes, and pseudonym exchange times, while we also examined how the energy of the nodes (vehicles) was affected when executing each asymmetric protocol. As it is easily seen from the various metrics, ECC is the most efficient when key generation takes place, and is very fast in decryption, but a bit slower in encryption. It is faster than RSA, for about 30% in encryption and decryption average time for 128-bit security level. A modern device for encrypting/decrypting messages using AES or modified ECC exchanged in the ISM band, was demonstrated, using low-cost commercial electronics. Hyperelliptic Curve Cryptography is not mature enough to supersede ECC, as it is shown by many metrics. In most metrics ECC is faster in times rather than HECC. This occurs because there is no optimized code for HECC genus 2 and HECC genus 3 as appears in ECC, which has been operating for many decades in various fields, apart from IoV. However, in energy consumption of specific messages HECC genus 3 seems to perform better than ECC.

There was a pioneer patented invention **Figure 146** regarding a mailbox for hardcopy letters. The user gets informed when a letter is received in their mailbox, via an SMS message in their mobile phones. The electronic mailbox consists of sensors that detect the reception of a new letter and via the use of Arduino MEGA 2560 R3 microcontroller is able to send an informative message to user. The device incorporates also a keypad in order to check some of its capabilities such as the change of the mobile number, check concerning the GSM/GPRS signal strength, information regarding the current/voltage/power of the device.

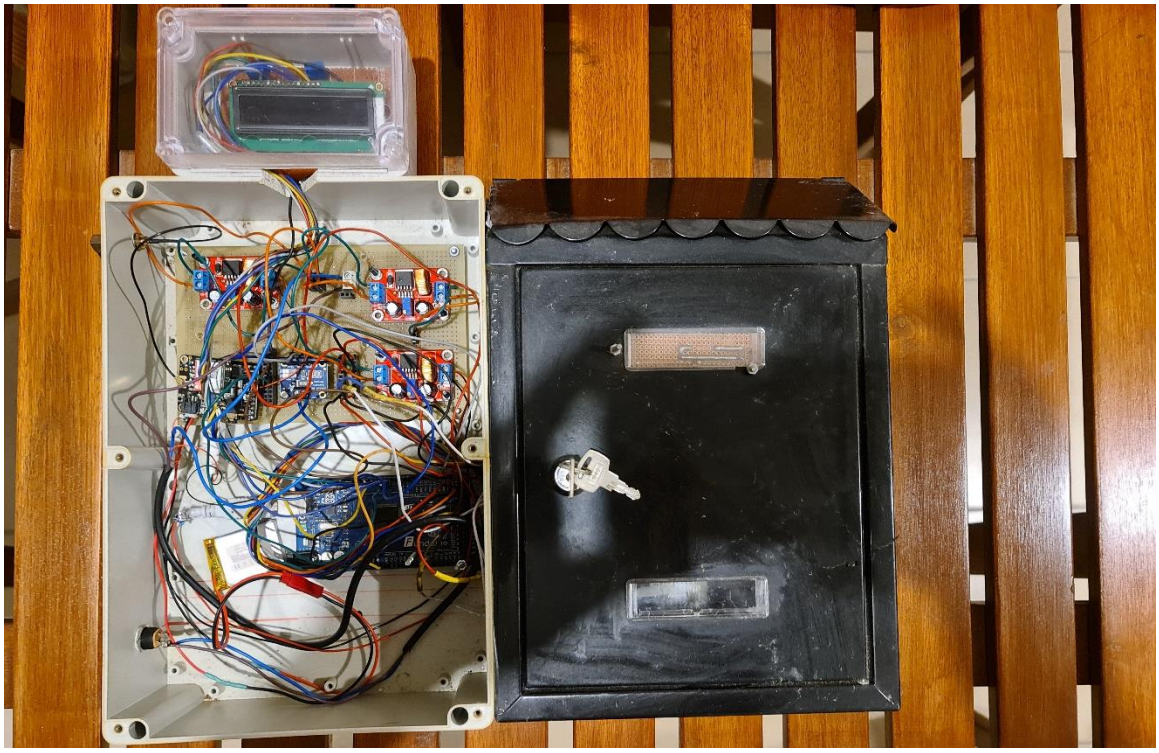


Figure 146 The patented electronic mailbox. On the left side someone can see the processing unit, whereas on the right side can see the mailbox which contains inside various sensors.

The Internet of Things keeps evolving in a fast pace and the lessons we have learnt are already being put to good use. The device presented in Chapter 1 will be scaled for usage in a real farm (northern Greece), sensing soil moisture, UV radiance, air temperature and humidity and transmitting the data to the Cloud, in order to predict future environmental parameters via the RNN-LSTM model and inform the user about the fore coming events on a local level. Secondly, a device that will monitor the leaves on a farm, identify diseases, and send the data to a ground station for further analysis is being built based on the experiments of Chapter 2. The ground station will consist of a specific hardware, such a GPU-based or TPU-based SBC, in order to fast process a lot of information and inform the end user timely about the condition of their field or potential dangers.

Our plan concerning the application in Chapter 3, is to improve it, in order to extend to different farm types and not only rice farm fields. The application was built with data coming from Rice farm field, and is optimized for soil salinity, but it can easily be extended to use different data from other yields and build different models that will be able to identify different parameters, such as pH, via only the use of UAV images.

In relation to Chapter 4, we use various and different methods in order to calculate the soil salinity in the Rice farm. We use VIs in the application and CNN ML models for estimating soil salinity in Rice fields. Also, we use RNN-LSTM for timeseries on Maize farm fields, that are near, but not IoT sensors placed inside the farm. The next step here is to combine all these applications into a web

service, while including a range of different crops that the farmers would choose according to their yield and the results will be sent straight to their smartphones.

Chapter 5, shows a new device capable of monitoring pine resin and environmental conditions, so that the pine resin collector is aware of what occurs in the pine forest. This device currently provides real-time data only through Zigbee modules and near real-time via SMS. We intend to use a 4G/5G cellular network to provide real-time data in every place the collector is, by building. a service running on Cloud via MQTT and showing information to the user with the cooperation of a 4G/5G data stream. The cost efficiency of this solution is a significant factor, so we will need to test different equipment and compare the results against the actual expected benefits and projected profit. Also, there are efforts to use more nodes that will communicate with the base node via Zigbee and the central node only will forward the data via GSM/GPRS/4G/5G to the end user. A illustrative image is depicted in **Figure 147**.

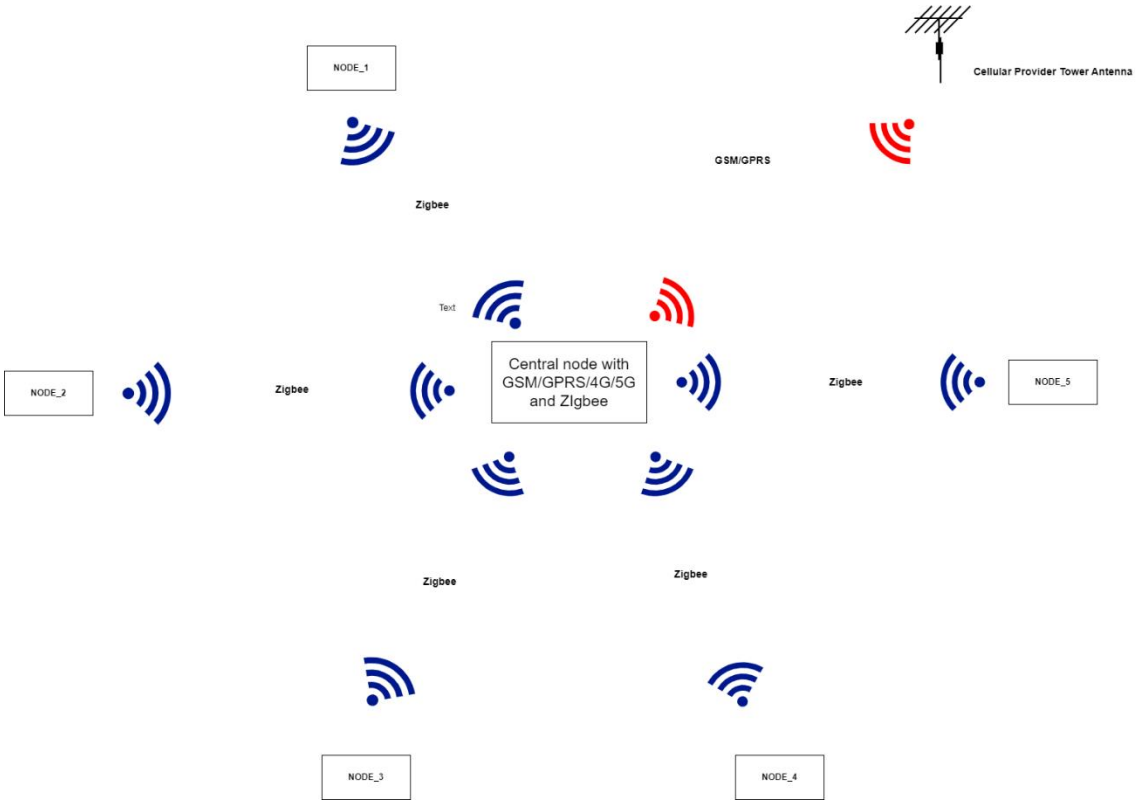


Figure 147 How the central node communicates with the rest nodes via Zigbee and the end-user via GSM/GPRS/4G/5G.

The model presented in Chapter 6**,** where we analyze the privacy in IoV using many asymmetric protocols, RSA, ECC, NTRU and symmetric protocols, AES, is ready to be tested with real vehicles and real RSUs. There are modules that can be connected to OBD-2 port on the cars and give output to a communication protocol that can be handled by Arduino. Thus, we could arrange a real-case scenario and monitor how each message and each protocol operates by taking various metrics, evaluating the whole model. We are currently waiting for the right call for proposals to turn this endeavour to an actual project.

Finally, we intend to optimize the HECC C++ code presented in Chapter 7, where again we analyze the IoV privacy using a different algorithm than Chapter 6 and different asymmetric protocols. Thus, we could get more precise metrics, because now the HECC C++ code used in both genus 2 and genus 3 is not mature enough. We believe this is why ECC shows better results than HECC, except for energy consumption in specific messages. It would be interesting to realize the whole construction in real cars, or vehicles, with real RSUs and see how they behave, but first we need to validate our assumptions about the algorithm's efficiency. If the results are positive, we are considering building a chip (ASIC or FPGA – Field Programmable Gate Array) specifically operating for those protocols (probably based on an implementation of HECC-2 or HECC-3 DLP encryption algorithm in VHDL) and making them fast enough to be used in the commercial sector.

The Internet of Things is everywhere and the underlying technologies are critical for anyone who wished to harness its full power. It is the combination of all the different experiments and studies we have performed that gives us the knowledge we need to stay in this fast-evolving world and be a part of the future.

This page was intentionally left blank.

List of author's publications

Journals:

1. **George Routis**, Marios Michailidis, Ioanna Roussaki, Plant disease identification using Machine Learning algorithms on, Single Board Computers in IoT environments, Electronics, 2024.

2. **Routis, George**, Dagas, Panagiotis and Roussaki, Ioanna, 2024. Enhancing Privacy in the Internet of Vehicles via Hyperelliptic Curve Cryptography. *Electronics*, *13*(4), p.730.

3. **George Routis**, George Katsouris, Ioanna Roussaki, *Cryptography-based Location Privacy Protection in the Internet of Vehicles*, Journal of Ambient Intelligence and Humanized Computing, Springer 2023 (in press)

4. **Routis, George**. and Roussaki, Ioanna., 2023. Low Power IoT Electronics in Precision Irrigation. *Smart Agricultural Technology*, *5*, p.100310.

5. Roussaki, Ioanna, Kevin Doolin, Antonio F. Gómez-Skarmeta, **George Routis**, Juan Antonio López-Morales, Ethel Claffey, Manuel Mora Tavarez and Juan Antonio Martínez. "Building an interoperable space for smart agriculture." *Digit. Commun. Networks* 9 (2022): 183-193.

6. Kalatzis, Nikos, **George Routis**, Yiorgos Marinellis, Marios Avgeris, Ioanna Roussaki, Symeon Papavassiliou and Miltiades E. Anagnostou. "Semantic Interoperability for IoT Platforms in Support of Decision Making: An Experiment on Early Wildfire Detection." *Sensors (Basel, Switzerland)* 19 (2019): n. page 528.

Conferences:

1. Nikos Kalatzis, Marios Paraskevopoulos, **Geoge Routis** and Ioanna Roussaki, "*Smart Farming data and IoT in support of agricultural policy monitoring"*, COINS Conference 2024, UK (submitted)

2. **Routis, George**, Marios Paraskevopoulos, Ioannis A. Vetsikas, Ioanna Roussaki, Dimitris G. Stavrakoudis and Dimitrios Katsantonis. "Data-Driven and Interoperable Smart Agriculture: An IoT-based Use-Case for Arable Crops." *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)* (2022): 1-8. Barcelona Spain.

3. Kalatzis, Nikos, **George Routis**, Ioanna Roussaki and Symeon Papavassiliou. "Enabling data interoperability for federated IoT experimentation infrastructures." *2018 Global Internet of Things Summit (GIoTS)* (2018): 1-6. Bilbao, Spain.

4. Roussaki, Ioanna, Pavlos Kosmides, **George Routis**, Kevin Doolin, Veronique Pevtschin and Angelo Marguglio. “A Multi-Actor Approach to promote the employment of IoT in Agriculture.” *2019 Global IoT Summit (GIoTS)* (2019): 1-6. Aarhus, Denmark.

Book Chapters:

1. Palma, R. *et al.* (2022). Agricultural Information Model.), Raul Palma, Ioanna Roussaki, Till Döhmen, Rob Atkinson, Soumya Brahma, Christoph Lange, **George Routis**, Marcin Plociennik & Szymon Mueller In: Bochtis, D.D., Sørensen, C.G., Fountas, S., Moysiadis, V., Pardalos, P.M. (eds Information and Communication Technologies for Agriculture—Theme III: Decision. Springer Optimization and Its Applications, vol 184. Springer, Cham. https://doi.org/10.1007/978-3-030-84152-2_1

Patents

1. Mailbox that informs users when receiving mail/envelope by sending SMS and using advanced control via the use of keypad.
   Patent number: (11):1010036
   Application patent number: (21):20200100592
   International classification:
   (51):IPC8: A47G 29/12
   IPC8: A47G 29/122
   IPC8: H04W 4/14

References

[1] Bhanu, K.N., Mahadevaswamy, H.S. and Jasmine, H.J., 2020, July. IoT based smart system for enhanced irrigation in agriculture. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 760-765). IEEE. https://doi.org/10.1109/ICESC48915.2020.9156026

[2] Ansari, S., Aslam, T., Ansari, A., Otero, P., Ahmed, I. and Maqbool, F., 2021. Internet of things: Technologies and applications. *Introduction to Internet of Things in Management Science and Operations Research: Implemented Studies*, pp.1-30. https://doi.org/10.1007/978-3-030-74644-5_1

[3] Reddy, K.S.P., Roopa, Y.M., LN, K.R. and Nandan, N.S., 2020, July. IoT based smart agriculture using machine learning. In *2020 Second international conference on inventive research in computing applications (ICIRCA)* (pp. 130-134). IEEE. https://doi.org/10.1109/ICIRCA48905.2020.9183373

[4] Ding, X. and Du, W., 2022, May. Drlic: Deep reinforcement learning for irrigation control. In *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)* (pp. 41-53). IEEE. https://doi.org/10.1109/IPSN54338.2022.00011

[5] Chen, H., Wei, Z., Zhang, B. and Peng, Z., 2022, July. Irrigation Scheduling Optimization for Ecological Security Water and Eco-Environment Relationship. In *2022 IEEE 12th International Conference on Electronics Information and Emergency Communication (ICEIEC)* (pp. 255-258). IEEE. https://doi.org/10.1109/ICEIEC54567.2022.9835044

[6] Farooq, M., Hussain, A., Hashim, S., Yang, L. and Ali, M., 2020, November. Automated Irrigation System based on irrigation gates using fuzzy logic. In *2020 International Conference on Internet of Things and Intelligent Applications (ITIA)* (pp. 1-5). IEEE. https://doi.org/10.1109/ITIA50152.2020.9312344

[7] Navarro, E., Costa, N. and Pereira, A., 2020. A systematic review of IoT solutions for smart farming. *Sensors*, *20*(15), p.4231. https://doi.org/10.3390/s20154231

[8] Farooq, M.S., Riaz, S., Abid, A., Abid, K. and Naeem, M.A., 2019. A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. *Ieee Access*, *7*, pp.156237-156271. https://doi.org/10.1109/ACCESS.2019.2949703

[9] Idoje, G., Dagiuklas, T. and Iqbal, M., 2021. Survey for smart farming technologies: Challenges and issues. *Computers & Electrical Engineering*, *92*, p.107104. https://doi.org/10.1016/j.compeleceng.2021.107104

[10] Ben Ayed, R. and Hanana, M., 2021. Artificial intelligence to improve the food and agriculture sector. *Journal of Food Quality*, *2021*, pp.1-7. https://doi.org/10.1155/2021/5584754

[11] Akhter, R. and Sofi, S.A., 2022. Precision agriculture using IoT data analytics and machine learning. *Journal of King Saud University-Computer and Information Sciences*, *34*(8), pp.5602-5618. https://doi.org/10.1016/j.jksuci.2021.05.013

[12] Thakare, S. and Bhagat, P.H., 2018, June. Arduino-based smart irrigation using sensors and ESP8266 WiFi module. In *2018 Second International Conference on intelligent computing and control systems (ICICCS)* (pp. 1-5). IEEE. https://doi.org/10.1109/ICCONS.2018.8663041

[13] Rajkumar, M.N., Abinaya, S. and Kumar, V.V., 2017, March. Intelligent irrigation system—An IOT based approach. In *2017 International Conference on Innovations in Green Energy and Healthcare Technologies (IGEHT)* (pp. 1-5). IEEE. https://doi.org/10.1109/IGEHT.2017.8094057

[14] Divyapriya, S., Vijayakumar, R., Ramkumar, M.S., Amudha, A., Nagaveni, P., Emayavaramban, G. and Mansoor, V., 2020, October. IoT Enabled Drip Irrigation System with Weather Forecasting. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 86-89). IEEE. https://doi.org/10.1109/I-SMAC49090.2020.9243349

[15] Senthilmurugan, M. and Chinnaiyan, R., 2021, January. IoT and machine learning based peer to peer platform for crop growth and disease monitoring system using blockchain. In *2021 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-5). IEEE. https://doi.org/10.1109/ICCCI50826.2021.9402435

[16] Ahmed, G.N. and Kamalakkannan, S., 2022, January. Micronutrient Classification in IoT Based Agriculture Using Machine Learning (ML) Algorithm. In *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 1-9). IEEE. https://doi.org/10.1109/ICSSIT53264.2022.9716293

[17] Lu, Y., An, J. and Shi, S., 2021, September. Research on smart agriculture iot system based heterogeneous networking technology. In *2021 IEEE 4th International Conference on Information Systems and Computer Aided Education (ICISCAE)* (pp. 485-488). IEEE. https://doi.org/10.1109/ICISCAE52414.2021.9590756

[18] Mohamed, A.T., Aly, H.H. and Little, T.A., 2021, October. A Comparative Study of Hourly Wind Speed and Power Forecasting Using Deep Learning Networks, Weka Time Series, and ARIMA Algorithms for Smart Grid Integration. In *2021 IEEE Electrical Power and Energy Conference (EPEC)* (pp. 273-278). IEEE . https://doi.org/10.1109/EPEC52095.2021.9621652

[19] Bellini, B., Becoña, J.P., Pereira, A.S., Vázquez, C. and Arnaud, A., 2019, May. IoT in the agribusiness, a power consumption view. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1-4). IEEE. https://doi.org/10.1109/ISCAS.2019.8702576

[20] Kökten, E., Çalışkan, B.C., Karamzadeh, S. and Soyak, E.G., 2020. Low-Power Agriculture IoT System with LoRa: Open Field Storage Observation. *Electrical, Control and Communication Engineering*, *16*(2), pp.88-94. https://doi.org/10.2478/ecce-2020-0013

[21] Alharbi, H.A. and Aldossary, M., 2021. Energy-efficient edge-fog-cloud architecture for IoT-based smart agriculture environment. *IEEE Access*, *9*, pp.110480-110492. https://doi.org/10.1109/ACCESS.2021.3101397

[22] Paller, G., Szármes, P. and Élo, G., 2015. Power consumption considerations of gsm-connected sensors in the agrodat. hu sensor network. *Sensors & Transducers*, *189*(6), p.52. https://www.sensorsportal.com/HTML/DIGEST/june_2015/Vol_189/P_2671.pdf

[23] Nandal, V. and Dahiya, S., 2021. IoT based energy-efficient data aggregation wireless sensor network in agriculture: a review. *PSYCHOLOGY AND EDUCATION*, *58*(1), pp.2985-3007. https://pdfs.semanticscholar.org/3abd/7ec8cfd374a733f129c1211cbaa3484604dc.pdf

[24] Rabka, M., Mariyanayagam, D. and Shukla, P., 2022. IoT-Based Horticulture Monitoring System. In *Intelligent Sustainable Systems: Selected Papers of WorldS4 2021, Volume 2* (pp. 765-774). Springer Singapore. https://doi.org/10.1007/978-981-16-6369-7_68

[25] Jiang, X., Zhang, H., Yi, E.A.B., Raghunathan, N., Mousoulis, C., Chaterji, S., Peroulis, D., Shakouri, A. and Bagchi, S., 2020. Hybrid low-power wide-area mesh network for IoT applications. *IEEE Internet of Things Journal*, *8*(2), pp.901-915. https://doi.org/10.1109/JIOT.2020.3009228

[26] Laksiri, H.G.C.R., Dharmagunawardhana, H.A.C. and Wijayakulasooriya, J.V., 2019, December. Design and optimization of IOT based smart irrigation system in Sri Lanka. In *2019 14th Conference on Industrial and Information Systems (ICIIS)* (pp. 198-202). IEEE. https://doi.org/10.1109/ICIIS47346.2019.9063272

[27] Ojo, M.O. and Zahid, A., 2022. Deep learning in controlled environment agriculture: A review of recent advancements, challenges and prospects. *Sensors*, *22*(20), p.7965. https://doi.org/10.3390/s22207965

[28] Hong, J., Lee, S., Lee, G., Yang, D., Bae, J.H., Kim, J., Kim, K. and Lim, K.J., 2021. Comparison of machine learning algorithms for discharge prediction of multipurpose dam. *Water*, *13*(23), p.3369. https://doi.org/10.3390/w13233369

[29] Chen, C.J., Li, Y.S., Tai, C.Y., Chen, Y.C. and Huang, Y.M., 2022. Pest incidence forecasting based on internet of things and long short-term memory network. *Applied Soft Computing*, *124*, p.108895. https://doi.org/10.1016/j.asoc.2022.108895

[30] Shadrin, D., Menshchikov, A., Somov, A., Bornemann, G., Hauslage, J. and Fedorov, M., 2019. Enabling precision agriculture through embedded sensing with artificial intelligence. *IEEE Transactions on Instrumentation and Measurement*, *69*(7), pp.4103-4113. https://doi.org/10.1109/TIM.2019.2947125

[31] Helsper, J.P., Ric de Vos, C.H., Maas, F.M., Jonker, H.H., Van Den Broeck, H.C., Jordi, W., Pot, C.S., Keizer, L.P. and Schapendonk, A.H., 2003. Response of selected antioxidants and pigments in tissues of Rosa hybrida and Fuchsia hybrida to supplemental UV-A exposure. Physiologia Plantarum, 117(2), pp.171-178. https://doi.org/10.1034/j.1399-3054.2003.00037.x

[32] Turcsányi, E. and Vass, I., 2000. Inhibition of Photosynthetic Electron Transport by UV-A Radiation Targets the Photosystem II Complex¶. *Photochemistry and Photobiology*, *72*(4), pp.513-520. https://doi.org/10.1562/0031-8655(2000)0720513IOPETB2.0.CO2

[33] Foyer, C.H., Lelandais, M. and Kunert, K.J., 1994. Photooxidative stress in plants. https://doi.org/10.1111/j.1399-3054.1994.tb03042.x

[34] Ehling-Schulz, M., Bilger, W. and Scherer, S., 1997. UV-B-induced synthesis of photoprotective pigments and extracellular polysaccharides in the terrestrial cyanobacterium Nostoc commune. *Journal of Bacteriology*, *179*(6), pp.1940-1945. https://doi.org/10.1128/jb.179.6.1940-1945.1997

[35] Döhler, G., 1998. Effect of UV radiation on pigments of the Antarctic macroalga Leptosomia simplex L. *Photosynthetica*, *35*, pp.473-476. https://doi.org/10.1023/A:1006932922895

[36] Jahnke, L.S., 1999. Massive carotenoid accumulation in Dunaliella bardawil induced by ultraviolet-A radiation. *Journal of Photochemistry and Photobiology B: Biology*, *48*(1), pp.68-74. https://doi.org/10.1016/S1011-1344(99)00012-3

[37] Lingakumar, K., Amudha, P. and Kulandaivelu, G., 1999. Exclusion of solar UV-B (280–315 nm) radiation on vegetative growth and photosynthetic activities in Vigna unguiculata L. *Plant Science*, *148*(2), pp.97-103. https://doi.org/10.1016/S0168-9452(99)00076-X

[38] Shiozaki, N., Hattori, I., Gojo, R. and Tezuka, T., 1999. Activation of growth and nodulation in a symbiotic system between pea plants and leguminous bacteria by near-UV radiation. *Journal of Photochemistry and Photobiology B: Biology*, *50*(1), pp.33-37. https://doi.org/10.1016/S1011-1344(99)00065-2

[39] Daza-Torres, M.C., Arias-Prado, P.C., Reyes-Trujillo, A. and Urrutia-Cobo, N., 2017. Basil (Ocimum basilicum L.) water needs calculated from the crop coefficient. *Ingeniería e Investigación*, *37*(3), pp.8-16.

[40] Allen, R.G., Pereira, L.S., Raes, D. and Smith, M., 1998. Crop evapotranspiration-Guidelines for computing crop water requirements-FAO Irrigation and drainage paper 56. *Fao, Rome*, *300*(9), p.D05109.

[41] Walters, K.J. and Currey, C.J., 2019. Growth and development of basil species in response to temperature. *HortScience*, *54*(11), pp.1915-1920. https://doi.org/10.21273/HORTSCI12976-18

[42] Moccaldi, L.A. and Runkle, E.S., 2007. Modeling the effects of temperature and photosynthetic daily light integral on growth and flowering of Salvia splendens and Tagetes patula. *Journal of the American Society for Horticultural Science*, *132*(3), pp.283-288. https://doi.org/10.21273/JASHS.132.3.283

[43] Blanchard M.G., Runkle, E.S. 2011. Temperature. In Ball redbook, Crop production, 18th ed.; J. Nau (ed.); Ball Publishing, West Chicago IL., Vol 2, p. 67–81.

[44] Roberts, E.H., 1987. Measurement and prediction of flowering in annual crops. *Manipulation of flowering.*, pp.17-50.

[45] Cohen, Y. and Ben-Naim, Y., 2016. Nocturnal fanning suppresses downy mildew epidemics in sweet basil. *PLoS One*, *11*(5), p.e0155330. https://doi.org/10.1371/journal.pone.0155330

[46] Luo, J., Ying, K. and Bai, J., 2005. Savitzky–Golay smoothing and differentiation filter for even number data. *Signal processing*, *85*(7), pp.1429-1434. https://doi.org/10.1016/j.sigpro.2005.02.002

[47] Wei, D., Wang, B., Lin, G., Liu, D., Dong, Z., Liu, H. and Liu, Y., 2017. Research on unstructured text data mining and fault classification based on RNN-LSTM with malfunction inspection report. *Energies*, *10*(3), p.406.

[48] Sahoo, B.B., Jha, R., Singh, A. and Kumar, D., 2019. Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting. *Acta Geophysica*, *67*(5), pp.1471-1481.

[49] Mishra, D., Khan, A., Tiwari, R. and Upadhay, S., 2018, February. Automated irrigation system-IoT based approach. In *2018 3rd International conference on internet of things: Smart Innovation and Usages (IoT-SIU)* (pp. 1-4). IEEE. https://doi.org/10.1109/IoT-SIU.2018.8519886

[50] Nalawade, R., Nagap, A., Jindam, L. and Ugale, M., 2020, April. Agriculture field monitoring and plant leaf disease detection. In *2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA)* (pp. 226-231). IEEE. https://doi.org/10.1109/CSCITA47329.2020.9137805

[51] Sarangdhar, A.A. and Pawar, V.R., 2017, April. Machine learning regression technique for cotton leaf disease detection and controlling using IoT. In *2017 international conference of electronics, communication and aerospace technology (ICECA)* (Vol. 2, pp. 449-454). IEEE. https://doi.org/10.1109/ICECA.2017.8212855

[52] Jiang, D., Li, F., Yang, Y. and Yu, S., 2020, August. A tomato leaf diseases classification method based on deep learning. In *2020 chinese control and decision conference (CCDC)* (pp. 1446-1450). IEEE. https://doi.org/10.1109/CCDC49329.2020.9164457

[53] Li, X. and Rai, L., 2020, November. Apple leaf disease identification and classification using resnet models. In *2020 IEEE 3rd International Conference on Electronic Information and Communication Technology (ICEICT)* (pp. 738-742). IEEE. https://doi.org/10.1109/ICEICT51264.2020.9334214

[54] Süzen, A.A., Duman, B. and Şen, B., 2020, June. Benchmark analysis of jetson tx2, jetson nano and raspberry pi using deep-cnn. In 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-5). IEEE.

[55] Liu, X., Cao, C. and Duan, S., 2023. A Low-Power Hardware Architecture for Real-Time CNN Computing. *Sensors*, *23*(4), p.2045.

[56] Zamir, M., Ali, N., Naseem, A., Ahmed Frasteen, A., Zafar, B., Assam, M., Othman, M. and Attia, E.A., 2022. Face detection & recognition from images & videos based on CNN & Raspberry Pi. Computation, 10(9), p.148.

[57] Monteiro, A., De Oliveira, M., De Oliveira, R. and Da Silva, T., 2018. Embedded application of convolutional neural networks on Raspberry Pi for SHM. Electronics Letters, 54(11), pp.680-682.

[58] Mittal, N. and Kumar, S., 2019, October. Machine Learning computation on multiple GPU's using CUDA and message passing interface. In 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC) (pp. 18-22). IEEE.

[59] Petrellis, N., 2017, May. A smart phone image processing application for plant disease diagnosis. In *2017 6th international conference on modern circuits and systems technologies (MOCAST)* (pp. 1-4). IEEE. https://doi.org/10.1109/MOCAST.2017.7937683

[60] Mittal, N. and Kumar, S., 2019, October. Machine Learning computation on multiple GPU's using CUDA and message passing interface. In *2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC)* (pp. 18-22). IEEE. https://doi.org/10.1109/PEEIC47157.2019.8976714

[61] Süzen, A.A., Duman, B. and Şen, B., 2020, June. Benchmark analysis of jetson tx2, jetson nano and raspberry pi using deep-cnn. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-5). IEEE. https://doi.org/10.1109/HORA49412.2020.9152915

[62] Kim, H., Nam, H., Jung, W. and Lee, J., 2017, April. Performance analysis of CNN frameworks for GPUs. In *2017 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)* (pp. 55-64). IEEE. https://doi.org/10.1109/ISPASS.2017.7975270

[63] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), pp.84-90. https://doi.org/10.1145/3065386

[64] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*, pp.211-252. https://doi.org/10.1007/s11263-015-0816-y

[65] Hara, K., Saito, D. and Shouno, H., 2015, July. Analysis of function of rectified linear unit used in deep learning. In *2015 international joint conference on neural networks (IJCNN)* (pp. 1-8). IEEE. https://doi.org/10.1109/IJCNN.2015.7280578

[66] Bottou, L., 1998. Online algorithms and stochastic approximations. *Online learning in neural networks*. https://cir.nii.ac.jp/crid/1570854175291941248

[67] Douarre, C., Schielein, R., Frindel, C., Gerth, S. and Rousseau, D., 2016. Deep learning based root-soil segmentation from X-ray tomography images. *BioRxiv*, p.071662. https://doi.org/10.1101/071662

[68] Chen, S.W., Shivakumar, S.S., Dcunha, S., Das, J., Okon, E., Qu, C., Taylor, C.J. and Kumar, V., 2017. Counting apples and oranges with deep learning: A data-driven approach. *IEEE Robotics and Automation Letters*, *2*(2), pp.781-788. https://doi.org/10.1109/LRA.2017.2651944

[69] Rahnemoonfar, M. and Sheppard, C., 2017. Deep count: fruit counting based on deep simulated learning. *Sensors*, *17*(4), p.905. https://doi.org/10.3390/s17040905

[70] Sarao, V., Veritti, D., Borrelli, E., Sadda, S.V.R., Poletti, E. and Lanzetta, P., 2019. A comparison between a white LED confocal imaging system and a conventional flash fundus camera using chromaticity analysis. *BMC ophthalmology*, *19*, pp.1-10. https://doi.org/10.1186/s12886-019-1241-8

[71] Mercado-Luna, A., Rico-García, E., Lara-Herrera, A., Soto-Zarazúa, G., Ocampo-Velázquez, R., Guevara-González, R., Herrera-Ruiz, G. and Torres-Pacheco, I., 2010. Nitrogen determination on tomato (Lycopersicon esculentum Mill.) seedlings by color image analysis (RGB). *African Journal of Biotechnology*, *9*(33). https://www.ajol.info/index.php/ajb/article/view/92074

[72] Moreira, L.C.J., Teixeira, A.D.S. and Galvão, L.S., 2015. Potential of multispectral and hyperspectral data to detect saline-exposed soils in Brazil. *GIScience & Remote Sensing*, *52*(4), pp.416-436. https://doi.org/10.1080/15481603.2015.1040227

[73] Nguyen, K.A., Liou, Y.A., Tran, H.P., Hoang, P.P. and Nguyen, T.H., 2020. Soil salinity assessment by using near-infrared channel and Vegetation Soil Salinity Index derived from Landsat 8 OLI data: a case study in the Tra Vinh Province, Mekong Delta, Vietnam. *Progress in Earth and Planetary Science*, *7*, pp.1-16. https://doi.org/10.1186/s40645-019-0311-0

[74] Tranmer, M., Murphy, J., Elliot, M., and Pampaka, M. (2020) Multiple Linear Regression (2nd Edition); Cathie Marsh Institute Working Paper 2020-01.

https://hummedia.manchester.ac.uk/institutes/cmist/archive-publications/working-papers/2020/2020-1-multiple-linear-regression.pdf

[75] Uyanık, G.K. and Güler, N., 2013. A study on multiple linear regression analysis. *Procedia-Social and Behavioral Sciences*, *106*, pp.234-240.

[76] Prats-Montalbán, J.M., de Juan, A. and Ferrer, A., 2011. Multivariate image analysis: A review with applications. *Chemometrics and intelligent laboratory systems*, *107*(1), pp.1-23. https://doi.org/10.1016/j.chemolab.2011.03.002

[77] Golabkesh, F., Ghanavati, N., Nazarpour, A. and Nejad, T.B., 2021. Monitoring Soil Salinity Changes, Comparison of Different Maps and Indices Extracted from Landsat Satellite Images (Case Study: Atabieh, Khuzestan). *Polish Journal of Environmental Studies*, *30*(2).

[78] Kumar, N., Reddy, G.O., Nagaraju, M.S.S. and Naitam, R.K., 2022. Remote sensing and machine learning for identification of salt-affected soils. Data Science in Agriculture and Natural Resource Management, pp.267-287.

[79] Wang, J., Peng, J., Li, H., Yin, C., Liu, W., Wang, T. and Zhang, H., 2021. Soil salinity mapping using machine learning algorithms with the Sentinel-2 MSI in arid areas, China. Remote Sensing, 13(2), p.305.

[80] Ivushkin, K., Bartholomeus, H., Bregt, A.K., Pulatov, A., Franceschini, M.H., Kramer, H., van Loo, E.N., Roman, V.J. and Finkers, R., 2019. UAV based soil salinity assessment of cropland. Geoderma, 338, pp.502-512.

[81] VS, F.E., 2020, May. Forecasting significant wave height using RNN-LSTM models. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1141-1146). IEEE.

[82] Ren, B., Xu, X. and Yu, H., 2021, October. Research of LSTM-RNN Model and Its Application Evaluation on Agricultural Products Circulation. In *2021 IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE)* (pp. 467-471). IEEE.

[83] Li, Y. and He, L., 2022, July. An Improved Object Detection CNN Module for Remote Sensing Images. In *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium* (pp. 1173-1176). IEEE.

[84] Fathabadi, F.R., Grantner, J.L., Abdel-Qader, I. and Shebrain, S.A., 2022. Box-trainer assessment system with real-time multi-class detection and tracking of laparoscopic instruments, using CNN. *Acta Polytechnica Hungarica*, *19*(2), pp.7-27.

[85] Yi, M., Jia, T., Dong, L., Zhang, L., Leng, C., Liu, S. and Lai, M., 2021. Resin yield in Pinus elliottii Engelm. is related to the resin flow rate, resin components and resin duct characteristics at three locations in southern China. *Industrial Crops and Products*, *160*, p.113141.

[86] Hadiyane, A., Sulistyawati, E., Asharina, W.P. and Dungani, R., 2015. A study on production of resin from Pinus merkusii Jungh. et de Vriese in the Bosscha observatory area, West Java-Indonesia. *Asian J. Plant Sci*, *14*(2), pp.89-93.

[87] Papajiannopoulos, A., 2002. Leaflet of resin tapping. *PRISMA Ltd. Athens, Greek*, pp.226-233.

[88] Gurau, V., Ragland, B., Cox, D., Michaud, A. and Busby, L., 2021. Robot Operations for Pine Tree Resin Collection. *Technologies*, *9*(4), p.79.

[89] Sharma, S.C., Prasad, N., Pandey, S.K. and Giri, S.K., 2018. Status of Resin tapping and scope of improvement: A review. *AMA Agric. Mech. Asia Afr. Lat. Am*, *49*, pp.16-26.

[90] Zevgolis, Y.G., Sazeides, C.I., Zannetos, S.P., Grammenou, V., Fyllas, N.M., Akriotis, T., Dimitrakopoulos, P.G. and Troumbis, A.Y., 2022. Investigating the effect of resin collection and detecting fungal infection in resin-tapped and non-tapped pine trees, using minimally invasive and non-invasive diagnostics. *Forest Ecology and Management*, *524*, p.120498.

[91] Haghshenas, G., Fard, F.R., Golmakani, M.T., Saharkhiz, M.J., Esmaeili, H., Khosravi, A.R. and Sedaghat, S., 2023. Yield, chemical composition, and antioxidant activity of essential oil obtained from Ferula persica oleo-gum-resin: Effect of the originated region, type of oleo-gum-resin, and extraction method. *Journal of Applied Research on Medicinal and Aromatic Plants*, *35*, p.100471.

[92] Nayak, A.P. and Prajapati, R.K., 2020. Sustainable gum tapping techniques for Lannea coromandelica (Houtt.) Merr. to obtain higher gum production in tropical dry deciduous forests. *Journal of Pharmacognosy and Phytochemistry*, *9*(2), pp.1347-1354.

[93] Fantin Irudaya Raj, E. and Appadurai, M., 2022. Internet of things-based smart transportation system for smart cities. In *Intelligent Systems for Social Good: Theory and Practice* (pp. 39-50). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-0770-8_4

[94] Patel, P., Narmawala, Z. and Thakkar, A., 2019. A survey on intelligent transportation system using internet of things. *Emerging Research in Computing, Information, Communication and Applications: ERCICA 2018, Volume 1*, pp.231-240. https://doi.org/10.1007/978-981-13-5953-8_20

[95] Ji, B., Zhang, X., Mumtaz, S., Han, C., Li, C., Wen, H. and Wang, D., 2020. Survey on the internet of vehicles: Network architectures and applications. *IEEE Communications Standards Magazine*, *4*(1), pp.34-41. https://doi.org/10.1109/MCOMSTD.001.1900053

[96] Qureshi, K.N., Din, S., Jeon, G. and Piccialli, F., 2020. Internet of vehicles: Key technologies, network model, solutions and challenges with future aspects. *IEEE Transactions on Intelligent Transportation Systems*, *22*(3), pp.1777-1786. https://doi.org/10.1109/TITS.2020.2994972

[97] Sharma, N., Chauhan, N. and Chand, N., 2018, December. Security challenges in Internet of Vehicles (IoV) environment. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)* (pp. 203-207). IEEE. https://doi.org/10.1109/ICSCCC.2018.8703272

[98] Malik, S. and Rana, A., 2022. Internet of Vehicles: Features, Architecture, Privacy, and Security Issues. In *Internet of Things: Security and Privacy in Cyberspace* (pp. 189-208). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-1585-7_9

[99] Khan, M.A. ed., 2022. *Internet of Things: A Hardware Development Perspective*. CRC Press. https://www.taylorfrancis.com/chapters/edit/10.1201/9781003122357-12/internetvehicles-abdullah-alharthi-qiang-ni-richard-jiang

[100] Hamdi, M.M., Audah, L., Rashid, S.A., Mohammed, A.H., Alani, S. and Mustafa, A.S., 2020, June. A review of applications, characteristics and challenges in vehicular ad hoc networks (VANETs). In *2020 international congress on human-computer interaction, optimization and robotic applications (HORA)* (pp. 1-7). IEEE. https://doi.org/10.1109/HORA49412.2020.9152928

[101] Eze, E.C., Zhang, S.J., Liu, E.J. and Eze, J.C., 2016. Advances in vehicular ad-hoc networks (VANETs): Challenges and road-map for future development. *International Journal of Automation and Computing*, *13*, pp.1-18. https://doi.org/10.1007/s11633-015-0913-y

[102] Shahwani, H., Shah, S.A., Ashraf, M., Akram, M., Jeong, J.P. and Shin, J., 2022. A comprehensive survey on data dissemination in Vehicular Ad Hoc Networks. *Vehicular Communications*, *34*, p.100420. https://doi.org/10.1016/j.vehcom.2021.100420

[103] Obaidat, M., Khodjaeva, M., Holst, J. and Ben Zid, M., 2020. Security and privacy challenges in vehicular ad hoc networks. *Connected Vehicles in the Internet of Things: Concepts, Technologies and Frameworks for the IoV*, pp.223-251. https://doi.org/10.1007/978-3-030-36167-9_9

[104] Kaur, R., Singh, T.P. and Khajuria, V., 2018, May. Security issues in vehicular ad-hoc network (VANET). In *2018 2nd International conference on trends in Electronics and Informatics (ICOEI)* (pp. 884-889). IEEE. https://doi.org/10.1109/ICOEI.2018.8553852

[105] Afzal, Z. and Kumar, M., 2020. Security of vehicular ad-hoc networks (VANET): a survey. In *Journal of Physics: Conference Series* (Vol. 1427, No. 1, p. 012015). IOP Publishing. https://iopscience.iop.org/article/10.1088/1742-6596/1427/1/012015/meta

[106] Zavvos, E., Gerding, E.H., Yazdanpanah, V., Maple, C. and Stein, S., 2021. Privacy and Trust in the Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), pp.10126-10141. https://doi.org/10.1109/TITS.2021.3121125

[107] Sun, Y., Wu, L., Wu, S., Li, S., Zhang, T., Zhang, L., Xu, J. and Xiong, Y., 2015, October. Security and Privacy in the Internet of Vehicles. In *2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI)* (pp. 116-121). IEEE. https://doi.org/10.1109/IIKI.2015.33

[108] Kumar, R., Kumar, P., Tripathi, R., Gupta, G.P. and Kumar, N., 2021. P2SF-IoV: A privacy-preservation-based secured framework for Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *23*(11), pp.22571-22582. https://doi.org/10.1109/TITS.2021.3102581

[109] Babaghayou, M., Labraoui, N., Abba Ari, A.A., Ferrag, M.A., Maglaras, L. and Janicke, H., 2021. Whisper: A location privacy-preserving scheme using transmission range changing for internet of vehicles. *Sensors*, *21*(7), p.2443. https://doi.org/10.3390/s21072443

[110] Kang, J., Yu, R., Huang, X., Jonsson, M., Bogucka, H., Gjessing, S. and Zhang, Y., 2016. Location privacy attacks and defenses in cloud-enabled internet of vehicles. *IEEE Wireless Communications*, *23*(5), pp.52-59. https://doi.org/10.1109/MWC.2016.7721742

[111] Yang, M., Feng, Y., Fu, X. and Qian, Q., 2019. Location privacy preserving scheme based on dynamic pseudonym swap zone for Internet of Vehicles. *International Journal of Distributed Sensor Networks*, *15*(7), p.1550147719865508. https://doi.org/10.1177/1550147719865508

[112] Zhang, J., Yang, F., Ma, Z., Wang, Z., Liu, X. and Ma, J., 2020. A decentralized location privacy-preserving spatial crowdsourcing for internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *22*(4), pp.2299-2313. https://doi.org/10.1109/TITS.2020.3010288

[113] Gupta, S. and Arora, G., 2019, November. Use of homomorphic encryption with GPS in location privacy. In *2019 4th International Conference on Information Systems and Computer Networks (ISCON)* (pp. 42-45). IEEE. https://doi.org/10.1109/ISCON47742.2019.9036149

[114] Benarous, L., Bitam, S. and Mellouk, A., 2021. CSLPPS: Concerted silence-based location privacy preserving scheme for internet of vehicles. *IEEE Transactions on Vehicular Technology*, *70*(7), pp.7153-7160. https://doi.org/10.1109/TVT.2021.3088762

[115] Huang, Q., Xu, X., Chen, H. and Xie, L., 2022. A vehicle trajectory privacy preservation method based on caching and dummy locations in the internet of vehicles. *Sensors*, *22*(12), p.4423. https://doi.org/10.3390/s22124423

[116] Kong, Q., Lu, R., Ma, M. and Bao, H., 2019. A privacy-preserving sensory data sharing scheme in Internet of Vehicles. *Future Generation Computer Systems*, *92*, pp.644-655. https://doi.org/10.1016/j.future.2017.12.003

[117] Kong, Q., Lu, R., Yin, F. and Cui, S., 2021. Blockchain-based privacy-preserving driver monitoring for MaaS in the vehicular IoT. *IEEE Transactions on Vehicular Technology*, *70*(4), pp.3788-3799. https://doi.org/10.1109/TVT.2021.3064834

[118] Mallouli, F., Hellal, A., Saeed, N.S. and Alzahrani, F.A., 2019, June. A survey on cryptography: comparative study between RSA vs ECC algorithms, and RSA vs El-Gamal algorithms. In *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)* (pp. 173-176). IEEE.

[119] Ma, M., 2021, October. Comparison between RSA and ECC. In *2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)* (pp. 642-645). IEEE.

[120] Chandel, S., Cao, W., Sun, Z., Yang, J., Zhang, B. and Ni, T.Y., 2020. A multi-dimensional adversary analysis of RSA and ECC in blockchain encryption. In *Advances in Information and Communication: Proceedings of the 2019 Future of Information and Communication Conference (FICC), Volume 2* (pp. 988-1003). Springer International Publishing.

[121] Paar, C. and Pelzl, J., 2009. *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media.

[122] Bafandehkar, M., Yasin, S.M., Mahmod, R. and Hanapi, Z.M., 2013, December. Comparison of ECC and RSA algorithm in resource constrained devices. In *2013 international conference on IT convergence and security (ICITCS)* (pp. 1-3). IEEE.

[123] Bansod, S. and Ragha, L., 2022, February. Secured and Quantum Resistant Key Exchange Cryptography Methods–A Comparison. In *2022 Interdisciplinary Research in Technology and Management (IRTM)* (pp. 1-5). IEEE.

[124] Agrawal, H. and Sharma, M., 2016. Calculation of complexity of NTRU and optimized NTRU using GA, ACO, and PSO algorithm. *Security and Communication Networks*, *9*(17), pp.4301-4318.

[125] Fernando, E., Agustin, D., Irsan, M., Murad, D.F., Rohayani, H. and Sujana, D., 2019, September. Performance comparison of symmetries encryption algorithm AES and DES with raspberry pi. In *2019 International Conference on Sustainable Information Engineering and Technology (SIET)* (pp. 353-357). IEEE.

[126] Orhanou, G., El Hajji, S. and Bentaleb, Y., 2011, April. EPS AES-based confidentiality and integrity algorithms: Complexity study. In *2011 International Conference on Multimedia Computing and Systems* (pp. 1-4). IEEE.

[127] Suárez-Albela, M., Fernández-Caramés, T.M., Fraga-Lamas, P. and Castedo, L., 2018, June. A practical performance comparison of ECC and RSA for resource-constrained IoT devices. In *2018 Global Internet of Things Summit (GIoTS)* (pp. 1-6). IEEE.

[128] Döring, R. and Geitz, M., 2022, April. Post-quantum cryptography in use: Empirical analysis of the TLS handshake performance. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium* (pp. 1-5). IEEE.

[129] Yaman, F., Mert, A.C., Öztürk, E. and Savaş, E., 2021, February. A hardware accelerator for polynomial multiplication operation of CRYSTALS-KYBER PQC scheme. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (pp. 1020-1025). IEEE.

[130] Yu, R., Kang, J., Huang, X., Xie, S., Zhang, Y. and Gjessing, S., 2015. MixGroup: Accumulative pseudonym exchanging for location privacy enhancement in vehicular social networks. *IEEE Transactions on Dependable and Secure Computing*, *13*(1), pp.93-105.

[131] Laksiri, H.G.C.R., Dharmagunawardhana, H.A.C. and Wijayakulasooriya, J.V., 2019, December. Design and optimization of IOT based smart irrigation system in Sri Lanka. In *2019 14th Conference on Industrial and Information Systems (ICIIS)* (pp. 198-202). IEEE.

[132] Babaghayou, M., Labraoui, N., Abba Ari, A.A., Ferrag, M.A., Maglaras, L. and Janicke, H., 2021. Whisper: A location privacy-preserving scheme using transmission range changing for internet of vehicles. *Sensors*, *21*(7), p.2443.

[133] Kang, J., Yu, R., Huang, X., Jonsson, M., Bogucka, H., Gjessing, S. and Zhang, Y., 2016. Location privacy attacks and defenses in cloud-enabled internet of vehicles. *IEEE Wireless Communications*, *23*(5), pp.52-59.

[134] Yang, M., Feng, Y., Fu, X. and Qian, Q., 2019. Location privacy preserving scheme based on dynamic pseudonym swap zone for Internet of Vehicles. *International Journal of Distributed Sensor Networks*, *15*(7), p.1550147719865508.

[135] Zhang, J., Yang, F., Ma, Z., Wang, Z., Liu, X. and Ma, J., 2020. A decentralized location privacy-preserving spatial crowdsourcing for internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *22*(4), pp.2299-2313.

[136] Gupta, S. and Arora, G., 2019, November. Use of homomorphic encryption with GPS in location privacy. In *2019 4th International Conference on Information Systems and Computer Networks (ISCON)* (pp. 42-45). IEEE.

[137] Benarous, L., Bitam, S. and Mellouk, A., 2021. CSLPPS: Concerted silence-based location privacy preserving scheme for internet of vehicles. *IEEE Transactions on Vehicular Technology*, *70*(7), pp.7153-7160.

[138] Huang, Q., Xu, X., Chen, H. and Xie, L., 2022. A vehicle trajectory privacy preservation method based on caching and dummy locations in the internet of vehicles. *Sensors*, *22*(12), p.4423.

[139] Kong, Q., Lu, R., Ma, M. and Bao, H., 2019. A privacy-preserving sensory data sharing scheme in Internet of Vehicles. *Future Generation Computer Systems*, *92*, pp.644-655.

[140] Kong, Q., Lu, R., Yin, F. and Cui, S., 2021. Blockchain-based privacy-preserving driver monitoring for MaaS in the vehicular IoT. *IEEE Transactions on Vehicular Technology*, *70*(4), pp.3788-3799.

[141] Mallouli, F., Hellal, A., Saeed, N.S. and Alzahrani, F.A., 2019, June. A survey on cryptography: comparative study between RSA vs ECC algorithms, and RSA vs El-Gamal algorithms. In *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)* (pp. 173-176). IEEE.

[142] Varfolomeev, A.A. and Makarov, A., 2020, January. About asymmetric execution of the asymmetric elgamal cipher. In *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)* (pp. 2106-2109). IEEE.

[143] Tu, Y., He, P., Koç, Ç.K. and Xie, J., 2023. LEAP: Lightweight and Efficient Accelerator for Sparse Polynomial Multiplication of HQC. IEEE Transactions on Very Large Scale Integration (VLSI) Systems.

[144] Chen, W., Wu, H., Chen, X. and Chen, J., 2022. A review of research on privacy protection of internet of vehicles based on blockchain. Journal of Sensor and Actuator Networks, 11(4), p.86.

[145] Khan, M.A., Ullah, I., Abdullah, A.M., Mohsan, S.A.H. and Noor, F., 2023. An efficient and conditional privacy-preserving heterogeneous signcryption scheme for the Internet of drones. Sensors, 23(3), p.1063.

[146] Shayea, G.G., Mohammed, D.A., Abbas, A.H. and Abdulsattar, N.F., 2022. Privacy-Aware Secure Routing through Elliptical Curve Cryptography with Optimal RSU Distribution in VANETs. Designs, 6(6), p.121.

[147] Al-Shareeda, M.A., Anbar, M., Manickam, S. and Hasbullah, I.H., 2022. A secure pseudonym-based conditional privacy-preservation authentication scheme in vehicular ad hoc networks. Sensors, 22(5), p.1696.

[148] Azees, M., Vijayakumar, P. and Deboarh, L.J., 2017. EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, *18*(9), pp.2467-2476.

[149] Liu, Y., Wang, Y. and Chang, G., 2017. Efficient privacy-preserving dual authentication and key agreement scheme for secure V2V communications in an IoV paradigm. *IEEE Transactions on Intelligent Transportation Systems*, *18*(10), pp.2740-2749.

[150] Jo, H.J., Kim, I.S. and Lee, D.H., 2017. Reliable cooperative authentication for vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, *19*(4), pp.1065-1079.

[151] Shen, Z., Wang, Y., Wang, H., Liu, P., Liu, K. and Zhang, J., 2024. Trust Mechanism Privacy Protection Scheme Combining Blockchain and Multi-Party Evaluation. IEEE Transactions on Intelligent Vehicles.

[152] Su, H., Dong, S., Wang, N. and Zhang, T., 2024. An efficient privacy-preserving authentication scheme that mitigates TA dependency in VANETs. Vehicular Communications, 45, p.100727.

[153] Mistareehi, H., Islam, T. and Manivannan, D., 2021. A secure and distributed architecture for vehicular cloud. *Internet of Things*, *13*, p.100355.

[154] Baee, M.A.R., Simpson, L., Boyen, X., Foo, E. and Pieprzyk, J., 2022. ALI: Anonymous lightweight inter-vehicle broadcast authentication with encryption. IEEE Transactions on Dependable and Secure Computing.

[155] Dua, A., Kumar, N., Das, A.K. and Susilo, W., 2017. Secure message communication protocol among vehicles in smart city. IEEE Transactions on Vehicular Technology, 67(5), pp.4359-4373.

[156] Mistareehi, H. and Manivannan, D., 2022. A low-overhead message authentication and secure message dissemination scheme for vanets. Network, 2(1), pp.139-152.

[157] Batool, H., Anjum, A., Khan, A., Izzo, S., Mazzocca, C. and Jeon, G., 2024. A secure and privacy preserved infrastructure for VANETs based on federated learning with local differential privacy. Information Sciences, 652, p.119717.

[158] Yadav, V.K., 2024. Anonymous and linkable ring signcryption scheme for location-based services in VANETs. Vehicular Communications, 45, p.100717.

[159] Shen, Z., Ren, F., Wang, H., Liu, P., Liu, K. and Zhang, J., 2024. Combining blockchain and crowd-sensing for location privacy protection in Internet of vehicles. Vehicular Communications, 45, p.100724.

[160] Huang, J., Kong, L., Wang, J., Chen, G., Gao, J., Huang, G. and Khan, M.K., 2024. Secure data sharing over vehicular networks based on multi-sharding blockchain. ACM Transactions on Sensor Networks, 20(2), pp.1-23.

[161] Tandon, R., Verma, A. and Gupta, P.K., 2024. D-BLAC: A dual blockchain-based decentralized architecture for authentication and communication in VANET. Expert Systems with Applications, 237, p.121461.

[162] Hammod, D.N., 2022, June. Modified Lightweight AES based on Replacement Table and Chaotic System. In 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-5). IEEE.

[163] Scheidler, R., 2015. An introduction to hyperelliptic curve arithmetic.

[164] Alimoradi, R., 2016. A study of hyperelliptic curves in cryptography. *International Journal of Computer Network and Information Security*, *8*(8), p.67.

[165] Mistareehi, H. and Manivannan, D., 2022. A low-overhead message authentication and secure message dissemination scheme for vanets. *Network*, *2*(1), pp.139-152.

[166] Bh, P., Chandravathi, D. and Roja, P.P., 2010. Encoding and decoding of a message in the implementation of Elliptic Curve cryptography using Koblitz's method. *International Journal on Computer Science and Engineering*, *2*(5), pp.1904-1907.

[167] Campagna, M., 2013. Standards for efficient cryptography sec 4: Elliptic curve QU-Vanstone implicit certificate Scheme (ECQV). *Certicom Corp*.

[168] Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K. and Vercauteren, F. eds., 2005. *Handbook of elliptic and hyperelliptic curve cryptography*. CRC press.

[169] Gaudry, P. and Schost, É., 2004. Construction of secure random curves of genus 2 over prime fields. In *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23* (pp. 239-256). Springer Berlin Heidelberg.

[170] Seck, M. and Diarra, N., 2018. Unified Formulas for Some Deterministic Almost-Injective Encodings into Hyperelliptic Curves. In *Progress in Cryptology–AFRICACRYPT 2018: 10th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 7–9, 2018, Proceedings 10* (pp. 183-202). Springer International Publishing.

[171] Weng, A., 2001. A class of hyperelliptic CM-curves of genus three. *Journal-Ramanujan Mathematical Society*, *16*(4), pp.339-372.

[172] Mistareehi, H. and Manivannan, D., 2022. A low-overhead message authentication and secure message dissemination scheme for vanets. Network, 2(1), pp.139-152.

[173] Cryptography, C., Elliptic and Hyperelliptic Curve Cryptography.

[174] Pelzl, J., Wollinger, T. and Paar, C., 2004. Special hyperelliptic curve cryptosystems of genus two: Efficient arithmetic and fast implementation. *Embedded Cryptographic Hardware: Design and Security*.

[175] Pelzl, J., Wollinger, T., Guajardo, J. and Paar, C., 2003. Hyperelliptic curve cryptosystems: Closing the performance gap to elliptic curves. In *Cryptographic Hardware and Embedded Systems-CHES 2003: 5th International Workshop, Cologne, Germany, September 8–10, 2003. Proceedings 5* (pp. 351-365). Springer Berlin Heidelberg.

[176] Seck, M. and Diarra, N., 2018. Unified Formulas for Some Deterministic Almost-Injective Encodings into Hyperelliptic Curves. In *Progress in Cryptology–AFRICACRYPT 2018: 10th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 7–9, 2018, Proceedings 10* (pp. 183-202). Springer International Publishing.

[177] Vuagnoux, M. and Pasini, S., 2009, August. Compromising electromagnetic emanations of wired and wireless keyboards. In *USENIX security symposium* (Vol. 8, pp. 1-16).

[178] Choi, H.J., Lee, H.S., Sim, D., Yook, J.G. and Sim, K., 2016, August. Reconstruction of leaked signal from USB keyboards. In *2016 URSI Asia-Pacific Radio Science Conference (URSI AP-RASC)* (pp. 1281-1283). IEEE.

[179] Vuagnoux, M. and Pasini, S., 2010, July. An improved technique to discover compromising electromagnetic emanations. In *2010 IEEE International Symposium on Electromagnetic Compatibility* (pp. 121-126). IEEE.

[180] Hu, J., Wang, H., Zheng, T., Hu, J., Chen, Z., Jiang, H. and Luo, J., 2023. Password-Stealing without Hacking: Wi-Fi Enabled Practical Keystroke Eavesdropping. *Proc. of the 30th ACM CCS*, pp.1-14.

[181] Války, G., 2012. RECONSTRUCTION OF MULTIPLE SIGNALS FROM ELECTROMAGNETIC COIL MEASUREMENT. *Journal of Electrical Engineering*, *63*(7s), pp.75-78.

[182] Choi, H.J., Lee, H.S., Sim, D., Yook, J.G. and Sim, K., 2016, August. Reconstruction of leaked signal from USB keyboards. In *2016 URSI Asia-Pacific Radio Science Conference (URSI AP-RASC)* (pp. 1281-1283). IEEE

[183] Babani, S., Bature, A.A., Faruk, M.I. and Dankadai, N.K., 2014. Comparative study between fiber optic and copper in communication link. *Int. J. Tech. Res. Appl*, *2*(2), pp.59-63.

[184] López-Cardona, J.D., Vázquez, C., Montero, D.S. and Lallana, P.C., 2017. Remote optical powering using fiber optics in hazardous environments. *Journal of Lightwave Technology*, *36*(3), pp.748-754.

[185] «Εφαρμογές IoT και Βελτιστοποίηση Διαχείρισης Πόρων στον τομέα της Ευφυούς Γεωργίας», Μάριος Μιχαηλίδης,

http://artemis.cslab.ece.ntua.gr:8080/jspui/bitstream/123456789/17789/1/%ce%94%ce%b9%cf%80%ce%bb%cf%89%ce%bc%ce%b1%cf%84%ce%b9%ce%ba%ce%b7%20%ce%95%cf%81%ce%b3%ce%b1%cf%83%ce%b9%ce%b1.pdf

[186] «Αξιοποίηση κρυπτογραφικών τεχνικών για τη βελτίωση της ιδιωτικότητας θέσης σε περιβάλλοντα IoV», Γεώργιος Κατσούρης, http://artemis.cslab.ece.ntua.gr:8080/jspui/bitstream/123456789/17934/1/Thesis_Katsouris_2021.pdf

[187] «Αξιοποίηση Κρυπτογραφικών Τεχνικών (Υπερ)Ελλειπτικών Καμπυλών για Βελτίωση της Ιδιωτικότητας σε Αυτό-Οργανούμενα Δίκτυα Οχημάτων», Παναγιώτης Ντάγκας, http://artemis.cslab.ece.ntua.gr:8080/jspui/bitstream/123456789/18729/1/Ntagkas_Thesis.pdf

This page was intentionally left blank.