



# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

## **Χρήση γνωσιακών πρακτόρων με επίγνωση πλαισίου για τη δημιουργία Κοινωνικού Διαδικτύου των Πραγμάτων**

Διδακτορική Διατριβή

**Παναγιώτης Κασνέσης**

Αθήνα, Φεβρουάριος 2018





# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

## Χρήση γνωσιακών πρακτόρων με επίγνωση πλαισίου για τη δημιουργία Κοινωνικού Διαδικτύου των Πραγμάτων

Διδακτορική Διατριβή

**Παναγιώτης Κασνέσης**

Συμβουλευτική Επιτροπή: Ιάκωβος Στ. Βενιέρης, Καθηγητής Ε.Μ.Π.

Δήμητρα-Θεοδώρα Ι. Κακλαμάνη, Καθηγήτρια Ε.Μ.Π.

Νικόλαος Κ. Ουζούνογλου, Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 8<sup>η</sup> Φεβρουαρίου 2018

.....  
Ι. Στ. Βενιέρης  
Καθηγητής Ε.Μ.Π.

.....  
Δ.-Θ. Ι. Κακλαμάνη  
Καθηγήτρια Ε.Μ.Π.

.....  
Ν. Κ. Ουζούνογλου  
Καθηγητής Ε.Μ.Π.

.....  
Α.-Γ. Σταφυλοπάτης,  
Καθηγητής Ε.Μ.Π.

.....  
Σ. Χρ. Παπαβασιλείου,  
Καθηγητής Ε.Μ.Π.

.....  
Γ. Ι. Στασινόπουλος  
Καθηγητής Ε.Μ.Π.

.....  
Χρ. Δουληγέρης,  
Καθηγητής ΠΑΠΕΙ

Αθήνα, Φεβρουάριος 2018

.....

**Παναγιώτης Κασνέσης**

Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Παναγιώτης Κασνέσης, 2018.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

*Σε αυτούς που με βελτιώνουν καθημερινά.*



## Περίληψη

Η έννοια του Διαδικτύου των Πραγμάτων (Internet of Things; IoT) είναι ήδη ώριμη, ώστε να ξεκινήσει να εξελίσσεται προς το Διαδίκτυο των Πάντων (Internet of Everything; IoE), μέσω του οποίου μπορούν να διασυνδεθούν άνθρωποι, αντικείμενα, ακόμη και εικονικά αντικείμενα ή εφαρμογές. Όμως για να καταστεί αυτό εφικτό, υπάρχει η ανάγκη για ένα πλαίσιο λογισμικού που θα στηρίζει όχι μόνο τη διασύνδεση μεταξύ των οντοτήτων του Διαδικτύου, αλλά θα επιτρέπει και την ουσιαστική αλληλεπίδραση μεταξύ τους. Συγκεκριμένα, οι διάφορες οντότητες μπορούν να ανταλλάσσουν δεδομένα και πληροφορίες έχοντας ως στόχο να καλύψουν τις ανάγκες των χρηστών τους. Λύση σε αυτό το πρόβλημα αποτελεί το Κοινωνικό Διαδίκτυο των Πραγμάτων (Social Internet of Things; SIoT), το οποίο αποτελεί σύγκλιση των Κοινωνικών Δικτύων και του Διαδικτύου των Πραγμάτων, και επιτρέπει το σχηματισμό σχέσεων φιλίας μεταξύ των αντικειμένων.

Στην παρούσα διδακτορική διατριβή παρουσιάζεται μία κατανεμημένη πλατφόρμα που αποτελείται από γνωσιακούς πράκτορες λογισμικού, οι οποίοι εκπροσωπούν τις οντότητες του IoE. Οι προτεινόμενοι πράκτορες λογισμικού έχουν ως στόχο την κάλυψη των ανθρώπινων αναγκών. Για αυτό το λόγο αποτελούνται από συστατικά στοιχεία λογισμικού ικανά να: (α) εξάγουν πληροφορίες πλαισίου από ακατέργαστα δεδομένα του IoT, και (β) να διαχειρίζονται γνωσιακά τις κοινωνικές τους σχέσεις με βάση τις ανάγκες τους, οι οποίες προκύπτουν από την ανάλυση των στόχων τους. Τα παραπάνω συστατικά στοιχεία λογισμικού βασίζονται αντιστοίχως σε: (α) ένα μοντέλο Βαθιάς Μάθησης (Deep Learning) για την αναγνώριση των ανθρώπινων δραστηριοτήτων, (β) έναν ευφυή κατανεμημένο μηχανισμό που κάνει χρήση των τεχνολογιών του Σημασιολογικού Ιστού και του προτύπου Δημοσίευσης-Συνδρομής (Publish-Subscribe pattern).

Για την υλοποίηση της προτεινόμενης λύσης για το Κοινωνικό Διαδίκτυο των Πραγμάτων, αναπτύχθηκαν δύο εργαλεία λογισμικού. Το πρώτο ονομάζεται Sem-SHUI και αποτελεί τη διεπαφή του χρήστη (δηλ., τον Πράκτορα Χρήστη) με το οικοσύστημα του SIoT, καθώς μπορεί να στείλει εντολές και να εισάγει σημασιολογικούς κανόνες που θα καθορίσουν τη συμπεριφορά των γνωσιακών πρακτόρων λογισμικού. Το δεύτερο είναι ένας προσομοιωτής του προτεινόμενου SIoT περιβάλλοντος και ονομάζεται ASSIST. Στόχος του ASSIST είναι να απεικονίσει γραφικά τις κοινωνικές σχέσεις των οντοτήτων του SIoT και να δώσει τη δυνατότητα στους προγραμματιστές να πειραματιστούν, αλλάζοντας τους κανόνες των πρακτόρων λογισμικού σχετικά με την ανακάλυψη υπηρεσιών και την πλοήγηση δικτύου.

Τέλος, για την αξιολόγηση της SIoT πλατφόρμας και των επιμέρους συστατικών της στοιχείων χρησιμοποιήθηκαν δεδομένα από αισθητήρες φορητών/κινητών

συσκευών, καθώς επίσης και τρία σενάρια χρήσης, τα οποία υποστηρίζονται από τα αναπτυχθέντα εργαλεία λογισμικού.

**Λέξεις κλειδιά:** κοινωνικό διαδίκτυο των πραγμάτων, γνωσιακοί πράκτορες λογισμικού, επίγνωση πλαισίου, σημασιολογικοί κανόνες, οντολογίες, βαθιά μάθηση, συνελκτικά νευρωνικά δίκτυα, αναγνώριση ανθρώπινης δραστηριότητας



## Abstract

The concept of an Internet of Things (IoT) is already mature, enough to start evolving towards an Internet of Everything (IoE), over which humans, objects and virtual items can be inter-connected. For this to be feasible though, there is the need for a framework which not only supports the interconnection between the entities of this Internet, but also allows the meaningful interaction between them. In particular, the smart IoE entities should be able to exchange data and information in order to fulfill their owners' needs. A solution to this problem constitutes Social Internet of Things (SIoT), which is the convergence of Social Networks and Internet of Things, and enables the creation of social relations between the smart objects.

This thesis introduces a distributed SIoT platform that consists of cognitive software agents, which represent the IoE entities. These software agents have as goal to fulfill human needs. As a result, they consist of multiple software components capable of: (a) extracting context information from raw data, and (b) managing cognitively the social relations based on their needs, which are produced by analyzing their goals. The aforementioned components are based, respectively, in the following state-of-the-art technologies: (a) a Deep Learning model for human activity recognition, and (b) a cognitive decentralized mechanism, which utilizes the Semantic Web technologies and the Publish-Subscribe pattern.

Furthermore, in order to materialize the proposed SIoT solution, two software tools were developed. The first one is called Sem-SHUI and constitutes the user's interface (e.g., the User Agent) with the SIoT ecosystem, since it is capable of sending direct orders or injecting semantic rules that will alter the behavior of the software agents. The second one is an agent-based SIoT simulator, called ASSIST, and displays the SIoT ecosystem, i.e., the existing SIoT Agents and all the formed relationships between them. Moreover, ASSIST enables the developers to change the semantic rules concerning the service discovery and the network navigability.

Finally, to test the proposed Deep Learning architecture, we used two public available HAR datasets, showing that the proposed model outperforms other state-of-the-art Deep Learning human activity recognition methods. The presented SIoT solution was examined by taking into consideration three use cases, utilizing the developed SIoT tools.

**Keywords:** social internet of things, cognitive software agents, context-awareness, semantic rules, ontologies, convolutional neural networks, human activity recognition



## Ευχαριστίες

Η παρούσα διδακτορική διατριβή αποτελεί το επιστέγασμα της ερευνητικής μου δραστηριότητας στο εργαστήριο Ευφύων Επικοινωνιών και Δικτύων Ευρείας Ζώνης του Ε.Μ.Π. Κατά τη διάρκεια αυτής της προσπάθειας, είχα την τιμή να γνωρίσω και να συνεργαστώ με αξιόλογους ανθρώπους, τους οποίους θα ήθελα να ευχαριστήσω για την πολύτιμη βοήθειά τους, καθώς και τη συνεχή στήριξη και καθοδήγηση που μου παρείχαν.

Καταρχήν, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κύριο Ιάκωβο Βενιέρη, Καθηγητή Ε.Μ.Π., για την ευκαιρία που μου έδωσε ώστε να μελετήσω ένα πολύ ενδιαφέρον ερευνητικό πεδίο, την πίστη του στις ικανότητες μου, αλλά και την άρτια καθοδήγηση που μου παρείχε, καθ' όλη τη διάρκεια εκπόνησης της διδακτορικής διατριβής μου. Θα ήθελα, επίσης, να ευχαριστήσω την κυρία Δήμητρα Κακλαμάνη, Καθηγήτρια Ε.Μ.Π., για την πολύτιμη καθοδήγηση και την υποστήριξή της όλα τα χρόνια της παρουσίας μου στο εργαστήριο. Πολλές ευχαριστίες οφείλω και στον κύριο Νικόλαο Ουζούνογλου, Καθηγητή Ε.Μ.Π., στον κύριο Ανδρέα-Γεώργιο Σταφυλοπάτη, Καθηγητή Ε.Μ.Π., και στον κύριο Συμεών Παπαβασιλείου, Καθηγητή Ε.Μ.Π., για τις εύστοχες υποδείξεις τους όσον αφορά στο κείμενο της ενδιάμεσης έκθεσης προόδου διατριβής. Επιπλέον, είμαι ευγνώμων στον κύριο Γεώργιο Στασινόπουλο, Καθηγητή Ε.Μ.Π., και στον κύριο Χρήστο Δουληγέρη, Καθηγητή του Πανεπιστημίου Πειραιώς, οι οποίοι δέχθηκαν να αποτελέσουν μέλη της επταμελούς επιτροπής μου. Ιδιαίτερη μνεία αξίζει ο κύριος Χαράλαμπος Πατρικάκης, Αν. Καθηγητής Α.Ε.Ι. Πειραιά Τ.Τ., ο οποίος με βοήθησε να εξελιχθώ επιστημονικά και η αρωγή του υπήρξε ανεκτίμητη, στην συγγραφή των επιστημονικών μου εργασιών.

Στη συνέχεια, θα ήθελα να ευχαριστήσω όλους τους συναδέλφους μου, τους οποίους, έχω, πλέον, τη χαρά να αποκαλώ φίλους. Συγκεκριμένα, ευχαριστώ τη Μαριέλα για την πολυετή συνεργασία μας και τις ατελείωτες «σπαγκοσυζητήσεις» μας· τον Αντρέα για την ανταλλαγή επιστημονικών και ποδοσφαιρικών απόψεων· τη Δέσποινα για τη σημαντική βοήθεια της· το Μανόλη για τον κώδικα που γράψαμε παρέα και για τις συζητήσεις μας περί ΑΕΚ· το Μιλανέζο Πέτρο για τις ποδοσφαιρικές αναλύσεις που κάναμε. Επίσης, ευχαριστώ τη Σοφία, τη Βάσω και τον Πάνο για την άψογη συνεργασία μας όλα αυτά τα χρόνια. Τέλος, είμαι ευγνώμων στο Λάζαρο για την βοήθεια του στην υλοποίηση του προσομοιωτή ASSIST, αλλά και στο Δημήτρη για την όμορφη συνεργασία που είχαμε.

Ένα τεράστιο «ευχαριστώ» οφείλω στη σύντροφο της ζωής μου Έφη, για τη συμπαράστασή της όλα αυτά τα χρόνια και για την κατανόησή της, ιδιαίτερα κατά την τελευταία περίοδο της προσπάθειάς μου. Ιδιαίτερα ευγνώμων είμαι στους γονείς μου, Κωνσταντίνο και Αγγελική Κασνέση για την ολόψυχη και ανιδιοτελή υποστήριξή

τους καθ' όλη τη διάρκεια της φοιτητικής ζωής μου, αλλά και στην αδερφή μου Κατερίνα για το άριστο *proofreading* που μου προσέφερε. Τέλος, θα ήθελα να ευχαριστήσω όλους τους φίλους μου και το αδερφό μου Γιώργο για τα γέλια που μου χάρισαν.

# Περιεχόμενα

Περίληψη .....	ii
Abstract .....	iv
Ευχαριστίες .....	vi
Περιεχόμενα .....	viii
Ευρετήριο Εικόνων .....	xii
Ευρετήριο Πινάκων.....	xvi
Πίνακας Μεταφράσεων Αγγλικών Όρων και Ακρωνυμίων.....	xviii
1 Εισαγωγή .....	1
1.1 Κίνητρα διδακτορικής διατριβής .....	2
1.1.1 Ανάγκη για δυναμική διασύνδεση μεταξύ των οντοτήτων του Διαδικτύου των Πραγμάτων .....	2
1.1.2 Ανάγκη για χρήση τεχνικών Βαθιάς Μάθησης με στόχο την επίγνωση πλαισίου. ....	4
1.2 Στόχος και συνεισφορά της διατριβής.....	5
1.3 Διάρθρωση της διατριβής.....	9
2 Θεωρητικό Υπόβαθρο .....	11
2.1 Διαδίκτυο των Πραγμάτων .....	11
2.1.1 Περιπτώσεις χρήσης του Διαδικτύου των Πραγμάτων.....	12
2.1.1.1 Παγκόσμιος Ιστός των Πραγμάτων .....	13
2.1.1.2 Κοινωνικό Διαδίκτυο των Πραγμάτων .....	14
2.1.1.3 Γνωσιακό Διαδίκτυο των Πραγμάτων .....	17
2.2 Υποστηρικτικές τεχνολογίες αιχμής για τη διασύνδεση των έξυπνων αντικειμένων.....	18
2.2.1 Πρότυπο ανταλλαγής μηνυμάτων Δημοσίευσης-Συνδρομής.....	18
2.2.2 Τεχνολογίες Σημασιολογικού Ιστού .....	20
2.2.3 Συλλογισμός σε οντολογικά δεδομένα .....	22
2.2.4 Μηχανισμοί ανίχνευσης συγκρούσεων σε συστήματα αυτοματισμού.....	27
2.3 Πολυπρακτορικά Συστήματα .....	28
2.4 Επίγνωση Πλαισίου .....	29
2.4.1 Ορισμός Επίγνωσης Πλαισίου .....	30
2.4.2 Εφαρμογές επίγνωσης πλαισίου με χρήση φορετών συσκευών.....	32
2.4.3 Στάδια Εξαγωγής και Επεξεργασίας Πληροφοριών Πλαισίου .....	35
2.5 Επεξεργασία Σημάτων Κίνησης .....	38

2.5.1	Συλλογή ακατέργαστων δεδομένων .....	39
2.5.2	Προεπεξεργασία δεδομένων .....	39
2.5.3	Τμηματοποίηση .....	39
2.5.4	Εξαγωγή Χαρακτηριστικών .....	40
2.5.5	Ταξινόμηση .....	41
2.6	Βαθιά Μάθηση.....	41
2.7	Συνελικτικά Νευρωνικά Δίκτυα .....	44
2.7.1	Επίπεδο Συνέλιξης .....	45
2.7.2	Ενεργοποίηση ReLu .....	46
2.7.3	Επίπεδο υποδειγματοληψίας μέγιστης τιμής .....	46
2.7.4	Επίπεδο ολικής υποδειγματοληψίας μέσης τιμής.....	47
2.7.5	Συνάρτηση ενεργοποίησης Softmax .....	47
2.7.6	Επίπεδο απόρριψης συνδέσεων dropout .....	48
2.7.7	Συνάρτηση Κόστους.....	48
2.8	Αλγόριθμοι Βαθιάς Μάθησης για Αναγνώριση Ανθρώπινης Δραστηριότητας 49	
2.8.1	Αυτοκωδικοποιητές .....	49
2.8.2	Συνελικτικά Νευρωνικά Δίκτυα για Αναγνώριση Ανθρώπινης Δραστηριότητας .....	50
2.8.3	Βαθιά Μάθηση σε φασματογραφήματα .....	51
2.8.4	Επαναλαμβανόμενα Νευρωνικά Δίκτυα .....	51
3	Αρχιτεκτονική πλατφόρμας και γνωσιακοί πράκτορες για το Κοινωνικό Διαδίκτυο των Πραγμάτων .....	53
3.1	Προτεινόμενη Αρχιτεκτονική για το Κοινωνικό Διαδίκτυο των Πραγμάτων	53
3.2	Γνωσιακοί Πράκτορες για το Κοινωνικό Διαδίκτυο των Πραγμάτων.....	55
3.3	Συστατικά στοιχεία λογισμικού του πράκτορα του SIoT.....	56
3.4	Σημασιολογική Αναπαράσταση Σχέσεων .....	59
3.4.1	Οντολογία Core.....	61
3.4.2	Οντολογία SSOR-Ont.....	63
3.4.3	Οντολογία SONS-Ont .....	65
3.4.4	Οντολογία Rule .....	67
3.5	Διαδικασία Κοινωνικοποίησης Αντικειμένου .....	69
3.5.1	Κύκλος Ζωής του Έξυπνου Αντικειμένου .....	69

3.5.2	Μηχανισμός Δημοσίευσης-Συνδρομής για το Κοινωνικό Διαδίκτυο των Πραγμάτων .....	73
4	Αναπτυχθέντα εργαλεία λογισμικού για το Κοινωνικό Διαδίκτυο των Πραγμάτων	75
4.1	Sem-SHUI.....	75
4.1.1	Καρτέλα Δημιουργίας Στιγμιότυπων .....	75
4.1.2	Καρτέλα Δημιουργίας Κανόνων .....	76
4.1.3	Καρτέλα Κατάστασης Συσκευών .....	78
4.1.4	Μηχανισμός ανίχνευσης συγκρούσεων κανόνων.....	80
4.1.4.1	Λογική πρώτου βαθμού για τους σημασιολογικούς κανόνες .....	80
4.1.4.2	Ροή εργασιών αποθήκευσης κανόνων .....	82
4.1.5	Μηχανισμός δημιουργίας στόχων .....	86
4.2	Ο Προσομοιωτής ASSIST .....	88
5	Αναγνώριση Ανθρώπινης Δραστηριότητας με χρήση του μοντέλου PerceptionNet	93
5.1	Ρύθμιση Πειραμάτων.....	93
5.1.1	Σύνολα Δεδομένων .....	93
5.1.1.1	UCL.....	94
5.1.1.2	PAMAP2 .....	94
5.1.2	Μετρικές Απόδοσης.....	95
5.2	Αλγόριθμος Βελτιστοποίησης.....	96
5.3	Αποτελέσματα.....	97
5.3.1	Φάση Επικύρωσης .....	97
5.3.2	Αρχιτεκτονική Βαθέος Συνελικτικού Νευρωνικού Δικτύου .....	100
5.3.3	Φάση δοκιμής για το σύνολο δεδομένων UCL.....	100
5.3.4	Φάση δοκιμής για το σύνολο δεδομένων PAMAP2.....	103
6	Περιπτώσεις χρήσης του Κοινωνικού Διαδικτύου των Πραγμάτων για τον έλεγχο των προτεινόμενων γνωσιακών μηχανισμών .....	107
6.1	Περίπτωση χρήσης του Κοινωνικού Διαδικτύου των Πραγμάτων στην Πολιτιστική Κληρονομιά .....	107
6.2	Περίπτωση χρήσης του Κοινωνικού Διαδικτύου των Πραγμάτων στην Ευημερία.....	113
6.3	Περίπτωση χρήσης έξυπνου σπιτιού .....	117
6.3.1	Έλεγχος του μηχανισμού ανίχνευσης συγκρούσεων .....	117
6.3.2	Μέτρηση χρόνου απόκρισης του στοιχείου συλλογισμού .....	121

7	Συμπεράσματα και προτάσεις για μελλοντική έρευνα .....	123
7.1	Αξιολόγηση Αποτελεσμάτων .....	123
7.2	Υποθέσεις και Περιορισμοί.....	126
7.3	Μελλοντική Έρευνα .....	127
7.3.1	Διαχείριση Φιλίας.....	127
7.3.2	Διαχείριση Στόχων .....	130
7.3.2.1	Ανακάλυψη Υπηρεσίας.....	130
7.3.2.2	Λήψη Αποφάσεων.....	132
7.3.3	Τεχνικές Τεχνητής Νοημοσύνης .....	133
8	Αναφορές.....	135
	Παράρτημα .....	149
A.1	Παρουσίαση κώδικα για Αναγνώριση Ανθρώπινης Δραστηριότητας (PerceptionNet) .....	149
A.2	Γραφικές παραστάσεις των σημάτων κίνησης κατά τη διάρκεια επεξεργασίας τους.....	153
	Δημοσιεύσεις.....	157
	Διεθνή περιοδικά.....	157
	Κεφάλαια σε βιβλία .....	157
	Πρακτικά συνεδρίων.....	157



## Ευρετήριο Εικόνων

Εικόνα 1: Ταξινόμηση των οντοτήτων του IoT με βάση την υπολογιστική ισχύ και την πολυπλοκότητά τους [1].	2
Εικόνα 2: Διάγραμμα σύγκρισης του ρυθμού αύξησης του αριθμού των συσκευών IoT σε σχέση με τον παγκόσμιο πληθυσμό και τον αριθμό των ανθρώπων που είναι συνδεδεμένοι στο Διαδίκτυο [6].	3
Εικόνα 3: Παρουσίαση βημάτων μίας κοινής διαδικασίας επεξεργασίας σήματος και μίας βασισμένης σε Βαθιά Μάθηση [14].	5
Εικόνα 4: Οι τρεις τύποι πρακτόρων λογισμικού (Device Agent, Human Agent, Task Agent) για το IoE [19].	6
Εικόνα 5: Η εξέλιξη του Διαδικτύου σε πέντε φάσεις. Ξεκινά με τη σύνδεση δύο υπολογιστών και στη συνέχεια κινείται προς τη δημιουργία του Παγκόσμιου Ιστού με τη σύνδεση μεγάλου αριθμού υπολογιστών. Το Κινητό Διαδίκτυο προέκυψε συνδέοντας τις κινητές συσκευές στο Διαδίκτυο. Στη συνέχεια, οι άνθρωποι εντάχθηκαν στο Διαδίκτυο μέσω Κοινωνικών Δικτύων. Τέλος, κινείται προς το Διαδίκτυο των Πραγμάτων συνδέοντας καθημερινά αντικείμενα με το Διαδίκτυο [21].	11
Εικόνα 6: Τα διάφορα επίπεδα περιγραφής του WoT [1].	13
Εικόνα 7: Απεικόνιση της πρόσβασης μίας εφαρμογής σε έναν κατάλογο υπηρεσιών IoT [31].	14
Εικόνα 8: Αρχιτεκτονική της IBM για το ClOT [43].	18
Εικόνα 9: Απεικόνιση του μηχανισμού Pub-Sub σε IoT περιβάλλον [44].	19
Εικόνα 10: Εφαρμογή τεχνικών συλλογισμού στα σύνολα TBox και ABox.	24
Εικόνα 11: Αλληλεπίδραση ενός πράκτορα λογισμικού με το SIoT.	29
Εικόνα 12: Στάδια εξαγωγής και επεξεργασίας πληροφοριών πλαισίου	36
Εικόνα 13: Η συνήθης διαδικασία επεξεργασίας σήματος [14].	39
Εικόνα 14: Γραφική παράσταση δεδομένων τριαξονικού επιταχυνσιόμετρου χρησιμοποιώντας χρονικό παράθυρο ίσο με 2,54 δευτερόλεπτα.	40
Εικόνα 15: Venn διάγραμμα που παρουσιάζει ότι η Βαθιά Μάθηση είναι μία ερευνητική περιοχή της Μάθησης Χαρακτηριστικών, η οποία είναι ένα υποσύνολο της Μηχανικής Μάθησης, η οποία, και αυτή με τη σειρά της είναι ένα πεδίο της Τεχνητής Νοημοσύνης [94].	42
Εικόνα 16: Διαγράμματα ροής που παρουσιάζουν πως οι διάφορες διαδικασίες ενός συστήματος AI σχετίζονται μεταξύ τους στο πλαίσιο των διαφορετικών επιστημονικών κλάδων της AI. Τα σκιασμένα κουτιά υποδηλώνουν συστατικά στοιχεία που έχουν τη δυνατότητα να μαθαίνουν, από μόνα τους, κάνοντας χρήση των δεδομένων εισόδου [94].	43
Εικόνα 17: Αρχιτεκτονική Βαθέος Συνελκτικού Νευρωνικού Δικτύου για αναγνώριση ανθρώπινης δραστηριότητας χρησιμοποιώντας δεδομένα τριαξονικού επιταχυνσιόμετρου [14].	45
Εικόνα 18: Δυσδιάστατη συνέλιξη σε μία RGB εικόνα [100].	46
Εικόνα 19: Μονοδιάστατη (μόνο ως προς τον άξονα X) υποδειγματοληψία μέγιστης τιμής.	47
Εικόνα 20: Απεικόνιση απόρριψης συνδέσεων στο δεξί νευρωνικό δίκτυο (b), σε αντίθεση με το αριστερό (a) [103].	48
Εικόνα 21: Απεικόνιση Αυτοκωδικοποιητή [111].	50

Εικόνα 22: Απεικόνιση της αρχιτεκτονικής του μοντέλου Convolutional LSTM σε σήματα αισθητήρων κίνησης [121].	52
Εικόνα 23: Αρχιτεκτονική του SIoT οικοσυστήματος.	54
Εικόνα 24: Αρχιτεκτονική γνωσιακού πράκτορα επίγνωσης πλαισίου για το SIoT ....	56
Εικόνα 25: Τα συστατικά στοιχεία λογισμικού του SIoT Agent	57
Εικόνα 26: Η προτεινόμενη οντολογία για την επίτευξη γνωσιακών λειτουργιών στο Κοινωνικό Διαδίκτυο των Πραγμάτων.	60
Εικόνα 27: Παρουσίαση της οντολογίας Core.	62
Εικόνα 28: Το σχήμα της οντολογίας SSOR-Ont [44].	64
Εικόνα 29: Μετάβαση από το αρχικό SIoT στο SIoT πρότυπο βασισμένο σε έξυπνους πράκτορες λογισμικού [44].	65
Εικόνα 30: Το σχήμα της οντολογίας SONS-Ont [44].	66
Εικόνα 31: Απεικόνιση της οντολογίας Rule	68
Εικόνα 32: Κύκλος ζωής του έξυπνου αντικειμένου [44].	70
Εικόνα 33: Το <i>αρχείο δήλωσης</i> μίας φορητής συσκευής (activity tracker) σε μορφή turtle.	70
Εικόνα 34: Ερώτημα SPARQL που επιστρέφει όλες τις υπηρεσίες, τις ανάγκες, και τους γονεϊκούς φίλους μαζί με τους EPC κωδικούς τους, από το <i>αρχείο δήλωσης</i> μίας συσκευής [44].	71
Εικόνα 35: Το <i>αρχείο κατάστασης</i> μίας έξυπνης λάμπας (lamp) σε μορφή turtle. ....	73
Εικόνα 36: Η <i>Καρτέλα Δημιουργίας Στιγμιότυπων</i> της Sem-SHUI.	76
Εικόνα 37: Η <i>Καρτέλα Δημιουργίας Κανόνων</i> της Sem-SHUI	77
Εικόνα 38: Η <i>Καρτέλα Κατάστασης Συσκευών</i> της διεπαφής Sem-SHUI	80
Εικόνα 39: Ερώτημα SPARQL για την εύρεση μίας συσκευής που προσφέρει φως στο δωμάτιο Bedroom1.	80
Εικόνα 40: Παράδειγμα μετατροπής ενός κανόνα από τη διεπαφή Sem-SHUI σε οντολογικά στιγμιότυπα, και στη συνέχεια σε σημασιολογικό κανόνα.	83
Εικόνα 41: Ερώτημα SPARQL για την εύρεση των μη αντικρουόμενων κανόνων, βασισμένο στην " <i>Αρχή μη-σύγκρουσης</i> "	84
Εικόνα 42: BPMN διάγραμμα περιγραφής της ροής εργασιών κατά την εισαγωγή νέου κανόνα.	86
Εικόνα 43: Ερώτημα SPARQL του πράκτορα δράστη για την επιβεβαίωση πως οι παράμετροι ενός συγκεκριμένου κανόνα είναι φίλοι του και για την εξέταση αν είναι συνδρομητής στα απαραίτητα θέματα.	87
Εικόνα 44: Ερώτημα SPARQL του πράκτορα της παραμέτρου του κανόνα για την εύρεση του URI του topic για το οποίο ενδιαφέρεται ο πράκτορας λογισμικού του δράστη.	87
Εικόνα 45: Το Κοινωνικό Δίκτυο Πρακτόρων Λογισμικού [19].	89
Εικόνα 46: Screenshot που δείχνει την καρτέλα που απαιτεί την εισαγωγή της διεύθυνσης URL του MQTT Broker στο ASSIST.	90
Εικόνα 47: Το GUI του ASSIST κατά τη δημιουργία ενός νέου αντικειμένου.	91
Εικόνα 48: Τα αποτελέσματα ορθότητας της χρήσης δισδιάστατης συνέλιξης στο 1 <sup>ο</sup> , 2 <sup>ο</sup> και 3 <sup>ο</sup> επίπεδο συνέλιξης του UCL συνόλου επικύρωσης [136].	98
Εικόνα 49: Τα αποτελέσματα ορθότητας βασισμένα στη χρήση διαφορετικών τελευταίων κρυφών επιπέδων του UCL συνόλου επικύρωσης.	99
Εικόνα 50: Η t-SNE οπτικοποίηση του τελευταίου κρυφού επιπέδου στο μοντέλο PerceptionNet για τις έξι δραστηριότητες [136].	99

Εικόνα 51: Ορθότητα δοκιμής για τα μοντέλα PerceptionNet όσον αφορά το σύνολο δεδομένων UCL [136].	101
Εικόνα 52: Ψευδοκώδικας πιθανοτικής Συλλογικής Μάθησης.	101
Εικόνα 53: Ο πίνακας σύγκρισης του συλλογικού μοντέλου PerceptionNet για το σύνολο δεδομένων UCL [136].	102
Εικόνα 54: Αρχιτεκτονικές μοντέλων Βαθιάς Μάθησης για αναγνώριση ανθρώπινης δραστηριότητας. Η 1 <sup>η</sup> αφορά το μοντέλο Convolutional LSTM που παρουσιάζεται στην [121], η 2 <sup>η</sup> τη συνήθη CNN που παρουσιάζεται στην [113] και η 3 <sup>η</sup> αφορά την προσέγγιση PerceptionNet [136].	103
Εικόνα 55: Ο πίνακας σύγκρισης του συλλογικού μοντέλου PerceptionNet για το σύνολο δεδομένων PAMAP2 [136].	104
Εικόνα 56: Το GUI του ASSIST για την περίπτωση χρήσης της πολιτιστικής κληρονομιάς [19].	108
Εικόνα 57: Το GUI του JADE sniffer agent που παρουσιάζει τα μηνύματα που ανταλλάξαν οι SIoT agents στην περίπτωση χρήσης της πολιτιστικής κληρονομιάς, μετά τη δημιουργία του πράκτορα <i>AnDet_App</i> [19].	111
Εικόνα 58: Ο αριθμός των μηνυμάτων FIPA που ανταλλάχθηκαν κατά τη διαδικασία ανακάλυψης υπηρεσίας, κάνοντας χρήση του κανόνα <i>naiveInfluencer</i> (μπλε χρώμα) ή του κανόνα <i>flooding</i> (πράσινο χρώμα).	113
Εικόνα 59: Δύο screenshots που παρουσιάζουν το οικοσύστημα του SIoT για την περίπτωση χρήσης της ευημερίας. Στο αριστερό, το όριο του μέγιστου αριθμού σχέσεων ισούται με δέκα, ενώ στο δεξί, με τρία.	116
Εικόνα 60: Το GUI του JADE sniffer agent που παρουσιάζει τα μηνύματα που ανταλλάξαν οι SIoT agents στην περίπτωση χρήσης της ευημερίας, μετά τη δημιουργία του <i>fitnessApp</i> agent.	117
Εικόνα 61: Το προσομοιωμένο περιβάλλον έξυπνου σπιτιού με χρήση του προσομοιωτή <i>Siafu</i> .	118
Εικόνα 62: Προειδοποιητικό μήνυμα κατά την εύρεση αντικρουόμενων κανόνων.	121
Εικόνα 63: Χρόνοι απόκρισης παραγωγικού συλλογισμού σε σχέση με τον αριθμό ενεργών κανόνων ανά συσκευή.	122



## Ευρετήριο Πινάκων

Πίνακας 1: Αισθητήρες κινητών/φορητών συσκευών που χρησιμοποιούνται σε εφαρμογές επίγνωσης πλαισίου .....	34
Πίνακας 2: Τεχνικές εξαγωγής και επεξεργασίας πληροφοριών πλαισίου .....	37
Πίνακας 3: Κανόνες με βάση την τοποθεσία των αντικειμένων .....	78
Πίνακας 4: Διαφορετικοί τύποι ατόμου .....	81
Πίνακας 5: Παράδειγμα μη αντικρουόμενων κανόνων .....	85
Πίνακας 6: Πειραματική ρύθμιση των υπερ-παραμέτρων του συνόλου δεδομένων UCL [136].....	96
Πίνακας 7: Σύγκριση του μοντέλου PerceptionNet με άλλα μοντέλα τεχνολογιών αιχμής [136]......	101
Πίνακας 8: Ορθότητα δοκιμής ανά συμμετέχοντα του συλλογικού μοντέλου PerceptionNet [136]. .....	102
Πίνακας 9: Σύγκριση αποτελεσμάτων στο σύνολο δεδομένων δοκιμής PAMAP2 της μεθόδου PerceptionNet με άλλα μοντέλα τεχνολογιών αιχμής [136]. .....	104
Πίνακας 10: Οι οντότητες του SIoT οικοσυστήματος στο σενάριο χρήσης της πολιτιστικής κληρονομιάς [19]......	109
Πίνακας 11: Σημασιολογικοί κανόνες για τις γνωσιακές λειτουργίες των SIoT Agents [19]. .....	109
Πίνακας 12: Σημασιολογικοί κανόνες για την ανακάλυψη υπηρεσιών.....	112
Πίνακας 13: Οι οντότητες του SIoT στο σενάριο χρήσης της ευημερίας.....	114
Πίνακας 14: Οι ανάγκες και οι υπηρεσίες της οντότητας <i>fitnessApp</i> . .....	115
Πίνακας 15: Σύνολο κανόνων που έχουν ως δράστη το κλιματιστικό που βρίσκεται στο καθιστικό.....	119
Πίνακας 16: Παράμετροι πειράματος μέτρησης του χρόνου απόκρισης των μηχανισμών συλλογισμού .....	121
Πίνακας 17: Αντιμετώπιση προβλημάτων των επιλεχθεισών τεχνικών εξαγωγής και επεξεργασίας της πληροφορίας πλαισίου .....	125



## Πίνακας Μεταφράσεων Αγγλικών Όρων και Ακρωνυμίων

Αγγλικός Όρος	Ακρωνύμιο	Ελληνικός Όρος
Ad-Hoc Networks	-	Αυτοοργανούμενα Δίκτυα
Anomaly Detection	-	Ανίχνευση Ανωμαλιών
Application Programming Interface	API	Διεπαφή Προγραμματισμού Εφαρμογών
Artificial Intelligence	AI	Τεχνητή Νοημοσύνη
Artificial Neural Networks	ANNs	Τεχνητά Νευρωνικά Δίκτυα
Autoencoders	-	Αυτοκωδικοποιητές
Big Data	-	Δεδομένα Μεγάλης Κλίμακας
Cognitive Internet of Things	CloT	Γνωσιακό Διαδίκτυο των Πραγμάτων
Co-location Object Relationship	C-LOR	Σχέση Συνεγκατάστασης Αντικειμένου
Constraint Satisfaction Problems	CSPs	Προβλήματα Ικανοποίησης Περιορισμού
Context-Aware Computing	-	Υπολογισμός Επίγνωσης Πλαισίου
Convolutional Neural Networks	CNNs/ ConvNets	Συνελικτικά Νευρωνικά Δίκτυα
Co-work Object Relationship	C-WOR	Σχέση Συνεργασίας Αντικειμένου
Crowdsensing	-	Συλλογική Ανίχνευση Δεδομένων
Data Property	-	Ιδιότητα Δεδομένων
Decision Trees	DT	Δέντρα Αποφάσεων
Deep Learning	DL	Βαθιά Μάθηση
Deep Neural Network	DNN	Βαθύ Νευρωνικό Δίκτυο
Deep Reinforcement Learning	-	Βαθιά Ενισχυτική Μάθηση
Discrete Fourier Transform	DFT	Διακριτός Μετασχηματισμός Fourier
Electrocardiography	ECG	Ηλεκτροκαρδιογράφημα
Electronic Product Code	EPC	Ηλεκτρονικό Κωδικό Προϊόντος
Everything-as-a-Service	EaaS	Τα Πάντα ως Υπηρεσία
Flooding	-	Δρομολογήσεις δια Πλημμύρας

Foundation for Intelligent Physical Agents	FIPA	-
Graphical User Interface	GUI	Γραφικό Περιβάλλον Χρήστη
Hidden Markov Models	HMM	Κρυφά Μαρκοβιανά Μοντέλα
International Telecommunication Union	ITU	Διεθνής Ένωση Τηλεπικοινωνιών
Internet of Everything	IoE	Διαδίκτυο των Πάντων
Internet of Things	IoT	Διαδίκτυο των Πραγμάτων
k-Nearest Neighbors algorithm	k-NN	Αλγόριθμος k Πλησιέστερου Γείτονα
Linked Open Data	LOD	Ανοικτά Συνδεδεμένα Δεδομένα
Long Short-Term Memory	LSTM	Μακρο-Βραχυπρόθεσμη Μνήμη
Machine Learning	ML	Μηχανική Μάθηση
Markov Random Fields	-	Τυχαία Μαρκοβιανά Πεδία
Message Queuing Telemetry Transport	MQTT	-
Mobile Ad-Hoc Networks	MANET	Κινητά Αυτοοργανούμενα Δίκτυα
Mobile Computing	-	Κινητός Υπολογισμός
Multi-Agent Systems	MAS	Πολυπρακτορικά Συστήματα
Near Field Communication	NFC	-
Object Property	-	Ιδιότητα Αντικειμένου
One-Shot Learning	-	Μάθηση με μία Προσπάθεια
Outlier Analysis	-	Ανίχνευση Περιπτώσεων Άτυπης Συμπεριφοράς
Ownership Object Relationship	OOR	Σχέση Ιδιοκτησίας Αντικειμένου
Parental Object Relationship	POR	Γονική Σχέση Αντικειμένου
Peer-to-Peer	P2P	Περιβάλλοντα Ομότιμων Κόμβων
Pervasive Computing	-	Διάχυτος Υπολογισμός
Policy Gradient	-	Αλγόριθμος Κλίσης Πολιτικής
Publish-Subscribe pattern	Pub-Sub	Μοτίβο Δημοσίευσης-Συνδρομής
Quality of Data	QoD	Ποιότητα των Δεδομένων
Quality of Experience	QoE	Ποιότητα της Εμπειρίας
Quality of Information	QoI	Ποιότητα των Πληροφοριών



Radio Frequency Identification	RFID	-
Random Forests	RF	Τυχαία Δάση
Random Walkers	-	Τυχαίοι Περιπατητές
Recurrent Neural Networks	RNNs	Επαναλαμβανόμενα Νευρωνικά Δίκτυα
Reinforcement Learning	-	Ενισχυτική Μάθηση
Representation Learning	-	Μάθηση Χαρακτηριστικών
Resource Description Framework	RDF	Πλαίσιο Περιγραφής Πόρων
Restricted Boltzmann Machines	RBM	Περιορισμένες Μηχανές Boltzmann
Self-Taught Learning	-	Αυτοδίδακτη Μάθηση
Semantic Web	-	Σημασιολογικός Ιστός
Semantic Web of Things	SWoT	Σημασιολογικός Ιστός των Πραγμάτων
Semi-Supervised Learning	-	Ημι-Επιβλεπόμενη Μάθηση
Siamese Neural Networks	-	Σιαμαία Νευρωνικά Δίκτυα
Smart Object's Needs and Services Ontology	SONS-Ont	Οντολογία Περιγραφής των Αναγκών και των Υπηρεσιών του Έξυπνου Αντικειμένου
Social Internet of Things	SIoT	Κοινωνικό Διαδίκτυο των Πραγμάτων
Social Networks	SNs	Κοινωνικά Δίκτυα
Social Object Relationship	SOR	Κοινωνική Σχέση Αντικειμένου
Social Smart Object's Relationships Ontology	SSOR-Ont	Οντολογία Περιγραφής των Σχέσεων του Κοινωνικού Έξυπνου Αντικειμένου
Sparse Autoencoder	-	Αραιά Συνδεδεμένο Αυτοκωδικοποιητή
Supervised Learning	-	Επιβλεπόμενη Μάθηση
Support Vector Machines	SVM	Μηχανές Διανυσμάτων Υποστήριξης
Transmission Control Protocol/Internet Protocol	TCP/IP	Πρωτόκολλο Ελέγχου Μετάδοσης/Πρωτόκολλο Διαδικτύου
Uniform Resource Identifier	URI	Μοναδικό Αναγνωριστικό Πόρου
Unmanned Aerial vehicles	UAVs	Μη Επανδρωμένα Αεροσκάφη
Unsupervised Learning	-	Μη-Επιβλεπόμενη Μάθηση

Vehicular Ad-Hoc Networks	VANET	Αυτοοργανούμενα Δίκτυα Οχημάτων
Web of Things	WoT	Παγκόσμιος Ιστός των Πραγμάτων
Web Ontology Language	OWL	Γλώσσα Οντολογίας Ιστού
Wireless Sensor Networks	WSNs	Ασύρματα Δίκτυα Αισθητήρων
World Wide Web	WWW	Παγκόσμιος Ιστός

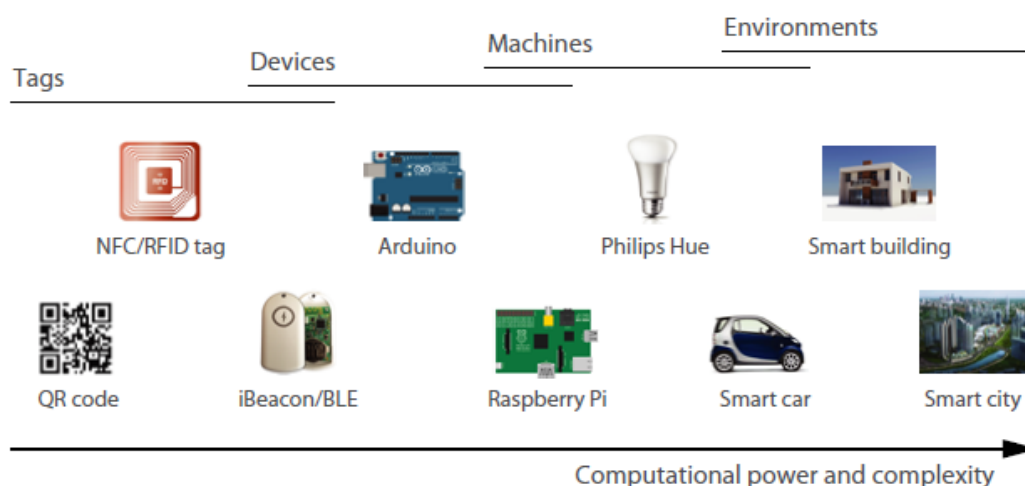
# 1 Εισαγωγή

Με το Διαδίκτυο να μετρά πάνω από 45 χρόνια ύπαρξης, κατά τη διάρκεια των οποίων άλλαξε ο τρόπος επικοινωνίας, ενημέρωσης και ψυχαγωγίας, το επόμενο βήμα στην εξέλιξή του είναι η επέκτασή του πέρα από την ανθρώπινη επικοινωνία. Προς αυτή την κατεύθυνση, οι μηχανές που μέχρι πρόσφατα χρησιμοποιήθηκαν για την επικοινωνία μέσω του Διαδικτύου συμμετέχουν ενεργά στη συγκεκριμένη επικοινωνία, επεκτείνοντας το παραδοσιακό Διαδίκτυο στο Διαδίκτυο των Πραγμάτων (Internet of Things; IoT), μέσω του οποίου οι άνθρωποι και οι μηχανές (συνδεδεμένα αντικείμενα) είναι σε θέση να επικοινωνούν. Σύμφωνα με την [1] το IoT είναι: *ένα σύστημα φυσικών αντικειμένων που μπορεί να ανακαλυφθεί, να παρακολουθηθεί, να ελεγχθεί ή να αλληλεπιδράσει με ηλεκτρονικές συσκευές που επικοινωνούν μέσω διαφόρων διασυνδέσεων δικτύωσης οι οποίες και τελικά μπορούν να συνδεθούν με το ευρύτερο διαδίκτυο.*

Κατά τη διάρκεια της τελευταίας δεκαετίας, το IoT έχει κερδίσει το ενδιαφέρον τόσο στον ακαδημαϊκό χώρο όσο και στη βιομηχανία. Οι κύριοι λόγοι πίσω από αυτό το ενδιαφέρον είναι οι δυνατότητες που προσφέρει το IoT [2]. Υπόσχεται να δημιουργήσει έναν κόσμο όπου όλα τα αντικείμενα (που ονομάζονται έξυπνα αντικείμενα [3]) συνδέονται με το Διαδίκτυο και επικοινωνούν μεταξύ τους με ελάχιστη ανθρώπινη παρέμβαση [4]. Ο απώτερος στόχος είναι να δημιουργηθεί ένας καλύτερος κόσμος για τα ανθρώπους, όπου τα αντικείμενα γύρω μας θα ξέρουν τι μας αρέσει, τι θέλουμε και τι χρειαζόμαστε και θα ενεργούν προς αυτή την κατεύθυνση χωρίς ρητές οδηγίες [5]. Ένα έξυπνο αντικείμενο, δηλαδή ένα αντικείμενο που ανήκει στο IoT, είναι ένα φυσικό αντικείμενο που ψηφιακά διαθέτει ένα ή περισσότερα από τα ακόλουθα:

- Αισθητήρες (π.χ., θερμοκρασίας, φωτεινότητας, κίνησης κλπ.)
- Ενεργοποιητές (π.χ., πρίζες κλπ.)
- Υπολογιστική ισχύ (δηλ., να μπορεί να τρέξει υπολογιστικά προγράμματα)
- Διεπαφές επικοινωνίας (ενσύρματες ή ασύρματες)

Συγκεκριμένα, οι οντότητες που ανήκουν στο IoT (Εικόνα 1) μπορεί να κυμαίνονται από απλά προϊόντα με ετικέτες RFID (Radio Frequency Identification) ή NFC (Near Field Communication), σε πιο πολύπλοκα και ασύρματα συνδεδεμένα προϊόντα, συσκευές (υπολογιστές μονής πλακέτας) ή μηχανήματα όπως τα συστήματα ασφαλείας και τα έξυπνα αυτοκίνητα, καταλήγοντας στην ύπαρξη έξυπνων κτιρίων ή ακόμα και στη δημιουργία μίας έξυπνης πόλης. Θα πρέπει να σημειωθεί πως απαραίτητη προϋπόθεση είναι η έμμεση (μέσω ραδιοκυμάτων μικρής εμβέλειας, όπως Bluetooth, ZigBee κτλ.) ή άμεση σύνδεση του αντικειμένου (ή τουλάχιστον των υπηρεσιών και των δεδομένων που παράγει) στο Διαδίκτυο.



Εικόνα 1: Ταξινόμηση των οντοτήτων του IoT με βάση την υπολογιστική ισχύ και την πολυπλοκότητά τους [1].

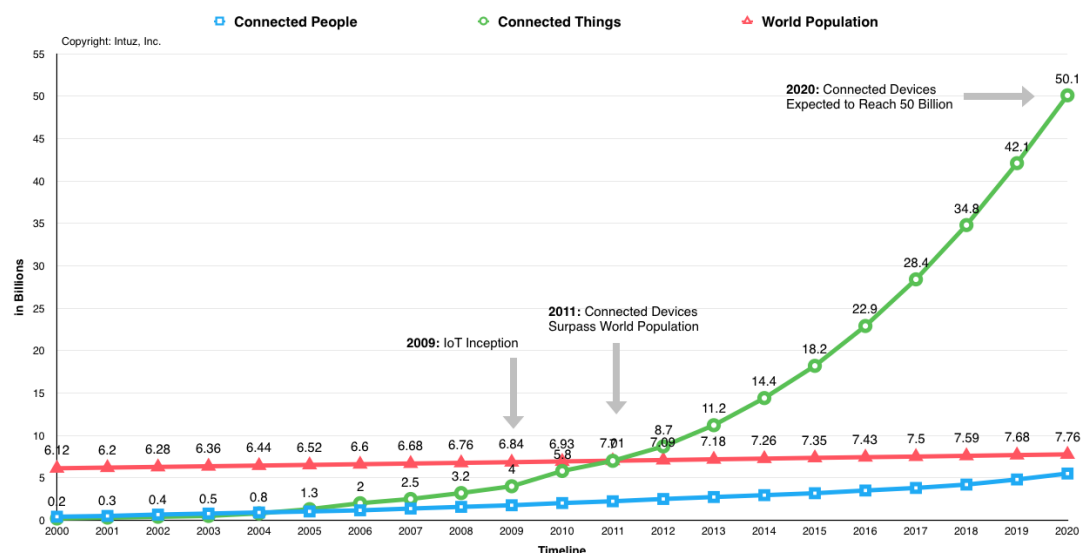
## 1.1 Κίνητρα διδακτορικής διατριβής

### 1.1.1 Ανάγκη για δυναμική διασύνδεση μεταξύ των οντοτήτων του Διαδικτύου των Πραγμάτων

Τα τελευταία χρόνια, το Διαδίκτυο των Πραγμάτων έχει γίνει μία από τις πιο ελπιδοφόρες και συναρπαστικές εξελίξεις στην τεχνολογία και τις επιχειρήσεις. Το όραμα ενός κόσμου, όπου μικροσκοπικοί υπολογιστές με αισθητήρες και διεπαφές επικοινωνίας που είναι ενσωματωμένοι στην υποδομή των πόλεών μας ή σε αυτοκίνητα, γραφεία ή ρούχα είναι πιθανό να φέρει επανάσταση σε κάθε τομέα της ζωής μας – στο παιχνίδι, την εργασία, και τον τρόπο διαβίωσής μας. Μέχρι πρόσφατα, οι πλατφόρμες IoT επικεντρώνονταν κατά κύριο λόγο στην οικοδόμηση εγκαταστάσεων μικρής κλίμακας, κλειστών και απομονωμένων, όπου οι συσκευές δεν σχεδιάζονται για εύκολη πρόσβαση ή επαναπρογραμματισμό. Η εξατομικευμένη σύζευξη μεταξύ συσκευών και εφαρμογών σε μία δεδομένη περίπτωση χρήσης, σημαίνει ότι οποιαδήποτε αλλαγή σε μία υπάρχουσα ανάπτυξη είναι πολύπλοκη και δαπανηρή. Αυτό περιορίζει τόσο τη συντήρηση όσο και την εξέλιξη του IoT, καθώς απαιτούνται σημαντικοί πόροι (χρόνος, χρήμα και τεχνικές δεξιότητες) κάθε φορά που προστίθεται μία νέα λειτουργία [1].

Σε αντίθεση, ο Παγκόσμιος Ιστός έχει καταστεί ευρέως επιτυχημένος τις τελευταίες δύο δεκαετίες, διότι είναι απλός στη χρήση και επίσης δίνει έμφαση στη χαλαρή σύνδεση μεταξύ διακομιστών, φυλλομετρητών ιστοσελίδων και εφαρμογών. Το απλό και σαφώς καθορισμένο μοντέλο προγραμματισμού του HTTP δίνει τη δυνατότητα σε οποιονδήποτε να αλλάξει κομμάτια του συστήματος, χωρίς να χρειαστεί να επαναπρογραμματίσει ολόκληρο το σύστημα. Επομένως, η οικοδόμηση νέων εφαρμογών ιστού ήταν σχετικά φθηνή και προσιτή σε μία πολύ μεγαλύτερη

ομάδα προγραμματιστών. Συνεπώς, έχει δοθεί ιδιαίτερη έμφαση στην ανάπτυξη πλατφορμών του IoT που έχουν ως κύριο στόχο τη μείωση του βάρους των προγραμματιστών του IoT και των παρόχων υπηρεσιών στην ανταλλαγή πληροφοριών, και τη διευκόλυνση της διαλειτουργικότητας μεταξύ συσκευών, δικτύων και τερματικών σημείων εξυπηρέτησης.



**Εικόνα 2: Διάγραμμα σύγκρισης του ρυθμού αύξησης του αριθμού των συσκευών IoT σε σχέση με τον παγκόσμιο πληθυσμό και τον αριθμό των ανθρώπων που είναι συνδεδεμένοι στο Διαδίκτυο [6].**

Ταυτόχρονα, λόγω της προόδου στην τεχνολογία των αισθητήρων, οι τελευταίοι γίνονται όλο και πιο ισχυροί, φθηνότεροι και μικρότεροι σε μέγεθος, γεγονός που ενθάρρυνε τις εφαρμογές μεγάλης κλίμακας. Ως αποτέλεσμα, σήμερα υπάρχει ένας μεγάλος αριθμός αισθητήρων που έχουν ήδη αναπτυχθεί και προβλέπεται ότι οι αριθμοί θα αυξηθούν ταχύτατα κατά την επόμενη δεκαετία [6]. Η Εικόνα 2 παρουσιάζει την ταχύτερη ανάπτυξη των αισθητήρων σε σχέση με τον ανθρώπινο πληθυσμό και το ποσοστό αυτών που χρησιμοποιούν το Διαδίκτυο. Ο αριθμός των συσκευών αναμένεται να φτάσει τα 50 δισεκατομμύρια μέχρι το 2020. Αυτές οι συσκευές θα παράγουν Δεδομένα Μεγάλης Κλίμακας (Big Data) [7], τα οποία δεν θα έχουν αξία εκτός αν αναλυθούν, ερμηνευθούν και κατανοηθούν. Ο Υπολογισμός Επίγνωσης Πλαισίου (Context-Aware Computing) έχει διαδραματίσει σημαντικό ρόλο στην αντιμετώπιση αυτής της πρόκλησης σε προηγούμενες περιπτώσεις χρήσης, όπως στον Κινητό και Διάχυτο Υπολογισμό (Mobile and Pervasive Computing), οι οποίες μας οδηγούν στο συμπέρασμα ότι θα συνεχίσει να είναι επιτυχής στην περίπτωση χρήσης του IoT. Ο Υπολογισμός Επίγνωσης Πλαισίου επιτρέπει την αποθήκευση πληροφοριών πλαισίου που συνδέονται με δεδομένα αισθητήρων, έτσι ώστε η ερμηνεία να γίνεται εύκολα και να έχει μεγαλύτερη σημασία. Επιπλέον, η κατανόηση του πλαισίου καθιστά ευκολότερη την επικοινωνία μεταξύ μηχανών, γεγονός που αποτελεί ένα βασικό στοιχείο στο όραμα του IoT.

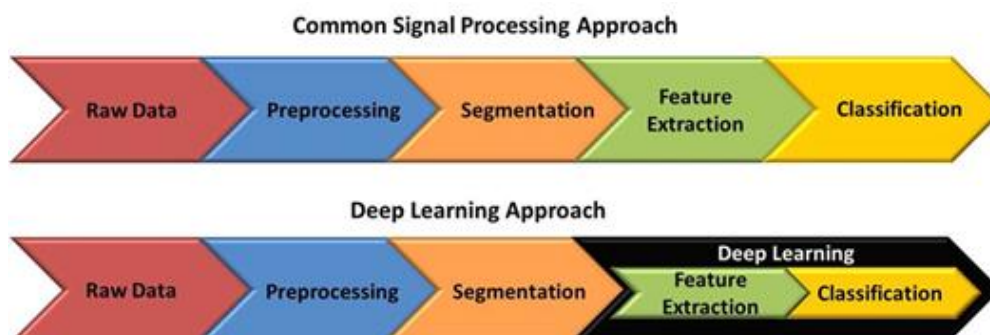
Τέλος, αν ληφθεί υπόψη ο πολλαπλασιασμός των Κοινωνικών Δικτύων (Social Networks; SNs) και η δυνατότητα που διαθέτουν για τη σύνδεση ανθρώπων με βάση τις προτιμήσεις και τα προσωπικά προφίλ τους, είναι φυσικό επακόλουθο η επικοινωνία μεταξύ των πραγμάτων (μέσω του Διαδικτύου) να υιοθετήσει και να επωφεληθεί από τις δυνατότητες κοινωνικοποίησης. Αλλά προτού να πραγματοποιηθεί μία βαθύτερη ανάλυση αυτής της δυνατότητας, η οποία θα συζητηθεί με λεπτομέρεια στο Κεφάλαιο 2, αξίζει μία πιο προσεκτική ματιά στον όρο Κοινωνικοποίηση: *Μία συνεχής διαδικασία μέσω της οποίας το άτομο αποκτά προσωπική ταυτότητα και μαθαίνει τους κανόνες, τις αξίες, τη συμπεριφορά και τις κοινωνικές δεξιότητες που είναι κατάλληλες για τη συμμετοχή του στην κοινωνία στην οποία ανήκει* [8]. Από τον ορισμό αυτό, μπορεί κανείς εύκολα να κατανοήσει τα οφέλη που μπορεί να εισαγάγει η κοινωνικοποίηση των συνδεδεμένων αντικειμένων στο IoT, καταλήγοντας στο σχηματισμό του Κοινωνικού Διαδικτύου των Πραγμάτων (Social Internet of Things; SIoT). Η ιδέα αυτή, η οποία αρχικά εισήχθη στο [9], παρέχει τη βάση πάνω στην οποία μπορεί να κατασκευαστεί με νόημα η αλληλεπίδραση μεταξύ των αντικειμένων (και των ανθρώπων).

### **1.1.2 Ανάγκη για χρήση τεχνικών Βαθιάς Μάθησης με στόχο την επίγνωση πλαισίου.**

Η δυνατότητα των συσκευών (φυσικών ή εικονικών) να κατανοούν, και να ερμηνεύουν το πλαίσιο τους ενισχύθηκε την τελευταία πενταετία, λόγω της άνθισης της Τεχνητής Νοημοσύνης (Artificial Intelligence; AI). Αλγόριθμοι και μοντέλα βασισμένοι στη χρήση παραγωγικής συλλογιστικής και τεχνικών Μηχανικής Μάθησης (Machine Learning; ML), άρχισαν να διαχειρίζονται τις λειτουργίες και τις συμπεριφορές των αντικείμενων, των ρομπότ και των πρακτόρων λογισμικού, καθιστώντας τες ολοένα και πιο έξυπνες. Επομένως, καθώς αυξάνεται ο αριθμός και η νοημοσύνη των έξυπνων αντικειμένων, υπάρχει ανάγκη να μετατραπούν οι διασυνδεδεμένοι κόμβοι του IoT σε αυτόνομες, συνεργατικές οντότητες που θα υποστηρίζουν τη δυνατότητα λήψης αποφάσεων χρησιμοποιώντας τόσο την τοπική όσο και την κατανεμημένη νοημοσύνη.

Πιο συγκεκριμένα, η στροφή προς την ανάπτυξη γνωσιακών εφαρμογών πυροδοτήθηκε από τις τεχνικές Βαθιάς Μάθησης (Deep Learning; DL) [10]. Η Βαθιά Μάθηση πρωτοεφαρμόστηκε στην υπολογιστική όραση, και απέδειξε ότι οι τεχνικές της επιτυγχάνουν καλύτερα αποτελέσματα από τις «παραδοσιακές» προσεγγίσεις που χρησιμοποιούν επιλεγμένα χαρακτηριστικά από ανθρώπους [11]. Ωστόσο, οι προσεγγίσεις Βαθιάς Μάθησης δεν περιορίζονται μόνο στην υπολογιστική όραση. Σχεδόν σε όλους τους τομείς που έχουν εφαρμοστεί (π.χ., αναγνώριση ομιλίας [12] και επεξεργασία φυσικής γλώσσας [13], προσφάτως και στην αναγνώριση ανθρώπινων δραστηριοτήτων (π.χ., τρέξιμο) έχουν καταφέρει να ξεπεράσουν τις τεχνικές που βασίζονται στα εξαχθέντα από ανθρώπους χαρακτηριστικά. Ένα από τα

μεγαλύτερα πλεονεκτήματα των Βαθέων Νευρωνικών Δικτύων (Deep Neural Networks; DNNs) είναι η ικανότητά τους να εξάγουν τα δικά τους χαρακτηριστικά, καθώς το βήμα της εξαγωγής χαρακτηριστικών περιλαμβάνεται στον αλγόριθμο DL [14][15]. Στην Εικόνα 3 παρουσιάζεται η προαναφερθείσα ιδιότητα τους.



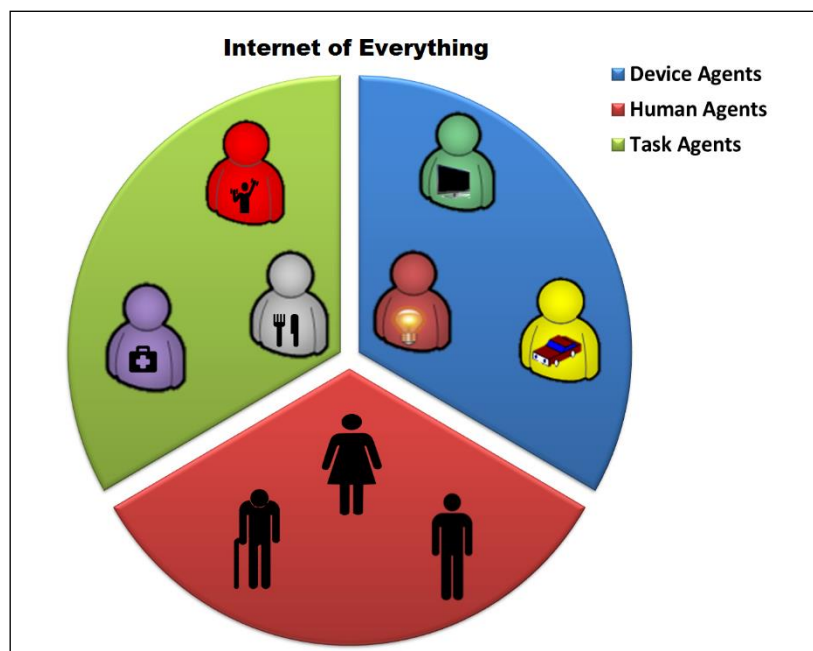
Εικόνα 3: Παρουσίαση βημάτων μίας κοινής διαδικασίας επεξεργασίας σήματος και μίας βασισμένης σε Βαθιά Μάθηση [14].

Παρόλα αυτά, η εφαρμογή τεχνικών Βαθιάς Μάθησης σε δεδομένα αισθητήρων δεν έχει γνωρίσει την απαραίτητη επιστημονική προσοχή και συνεπώς χρήζει περαιτέρω μελέτης. Ο πολλαπλασιασμός των συσκευών του IoT τα τελευταία χρόνια, και η τεράστια ποσότητα δεδομένων που παράγουν οι αισθητήρες (μία φορητή συσκευή παράγει εκατοντάδες δεδομένα κάθε δευτερόλεπτο) έχουν συμβάλει σε μία «έκρηξη» δεδομένων. Μέσω, λοιπόν, της επεξεργασίας των δεδομένων σε πραγματικό χρόνο ή σε παρτίδες, μπορεί να εξαχθούν σημαντικές πληροφορίες πλαισίου σχετικά με τα καθημερινά πρότυπα των ατόμων ή των κοινωνικών ομάδων που ανήκουν. Οι εν λόγω πληροφορίες οδηγούν στην ανάπτυξη εξατομικευμένων εφαρμογών, οι οποίες δύνανται να συμβάλουν στη βελτίωση της ποιότητας ζωής [16], στην εξασφάλιση καλύτερων συνθηκών υγείας [17], στην αύξηση του αισθήματος της ασφάλειας [18] κτλ.

## 1.2 Στόχος και συνεισφορά της διατριβής

Η παρούσα διατριβή επεκτείνει την έννοια των έξυπνων διασυνδεδεμένων συστημάτων, όπως αυτή ορίζεται στο IoT, επιτρέποντας τη σύνδεση και τη συνεργασία μεταξύ των οντοτήτων του κυβερνο-φυσικού χώρου (πράκτορες λογισμικού και εφαρμογές) και του φυσικού κόσμου (άνθρωποι και αντικείμενα). Αυτό οδηγεί στο σχηματισμό ενός κυβερνο-φυσικού οικοσυστήματος, στο οποίο οι άνθρωποι και τα πράγματα συνδέονται και αλληλεπιδρούν μέσω της χρήσης οντολογιών και ειδικών κανόνων. Ο λόγος πίσω από τη δημιουργία νέων σχέσεων μεταξύ των οντοτήτων είναι η ικανοποίηση του μεγάλου αριθμού των αναγκών που δηλώνουν αυτές οι οντότητες (κυρίως οι άνθρωποι). Καθώς ο αριθμός των σχέσεων μεταξύ των συσκευών αυξάνεται, η πιθανότητα ότι μία συγκεκριμένη ανάγκη μπορεί να εξυπηρετηθεί από έναν φίλο μίας οποιασδήποτε συσκευής στο δίκτυο αυξάνεται

αναλόγως, οδηγώντας σε έναν συνεργατικό κόσμο οντοτήτων που αποτελούν ένα πολυπρακτορικό σύστημα (Multi-Agent System; MAS).



Εικόνα 4: Οι τρεις τύποι πρακτόρων λογισμικού (Device Agent, Human Agent, Task Agent) για το IoE [19].

Το προτεινόμενο πολυπρακτορικό σύστημα είναι ένα περιβάλλον που περιλαμβάνει πολλούς αλληλεπιδραστικούς γνωσιακούς πράκτορες. Συγκεκριμένα, η παρούσα διδακτορική διατριβή ακολουθεί το πρότυπο του Διαδικτύου των Πάντων (Internet of Everything; IoE), το οποίο επεκτείνει το IoT μέσω της συμμετοχής των ανθρώπων και των εφαρμογών (υπηρεσίες και δεδομένα) στο δίκτυο. Η Εικόνα 4 παρουσιάζει τους τρεις τύπους πρακτόρων λογισμικού, δηλαδή τους Πράκτορες Συσκευών (Device Agents), τους Ανθρώπινους Πράκτορες (Human Agents) και τους Πράκτορες Εργασίας (Task Agents). Οι Πράκτορες Συσκευών αντιπροσωπεύουν τα έξυπνα αντικείμενα, οι Ανθρώπινοι Πράκτορες αντιπροσωπεύουν τους ανθρώπους και οι Πράκτορες Εργασίας αντιπροσωπεύουν τις εφαρμογές λογισμικού.

Όλες αυτές οι οντότητες παράγουν υπηρεσίες και έχουν ανάγκες. Για παράδειγμα, έστω ότι ένας Ανθρώπινος Πράκτορας έχει την ανάγκη να καθαρίσει το σπίτι του. Εάν υπάρχει ένας πράκτορας υπεύθυνος για τον καθαρισμό, τότε ο Ανθρώπινος Πράκτορας θα επικοινωνήσει με τους φίλους του που παράγουν υπηρεσίες καθαρισμού (π.χ., έξυπνη ηλεκτρική σκούπα) για να ολοκληρώσουν αυτό το έργο. Ωστόσο, οι Ανθρώπινοι Πράκτορες μπορούν να παράγουν και υπηρεσίες ενεργώντας ως αισθητήρες. Εάν ένας άνθρωπος ενοχληθεί από το θόρυβο που παράγει η ηλεκτρική σκούπα, μπορεί να ενεργήσει (μέσω της διεπαφής του αντίστοιχου Ανθρώπινου Πράκτορα) ως αισθητήρας θορύβου και να ενημερώσει τον πράκτορα καθαρισμού για αυτή την ενοχλητική κατάσταση. Σε αυτή την περίπτωση,



ο πράκτορας καθαρισμού μπορεί να επικοινωνήσει με τοπικούς αισθητήρες εντοπισμού θέσης στην προσπάθειά του να καθαρίσει τα δωμάτια όπου ο εν λόγω χρήστης δεν είναι παρών. Η επικοινωνία μεταξύ ανθρώπων, συσκευών και πρακτόρων θα πρέπει να ενεργοποιείται μέσω έξυπνων διεπαφών και ελαφρών πρωτοκόλλων IoT.

Ο κύριος στόχος της παρούσας διατριβής είναι η δημιουργία ενός γνωσιακού μηχανισμού ικανού να διαχειρίζεται τις κοινωνικές σχέσεις των έξυπνων οντοτήτων του SIoT, έχοντας ως επίκεντρο τις ανθρώπινες ανάγκες. Επιπλέον, ο συγκεκριμένος μηχανισμός θα πρέπει να μπορεί να αντιμετωπίσει τα θέματα διαλειτουργικότητας που σχετίζονται με τη διαδικασία ανακάλυψης υπηρεσιών, με σκοπό τη συνεργασία των έξυπνων φορέων για την επίτευξη κοινών στόχων. Τέλος, θα πρέπει να μπορεί να εξαγάγει ορθές πληροφορίες πλαισίου και να τις διαχειρίζεται με βάση τις εκάστοτε ανάγκες του χρήστη. Οι βασικές συνεισφορές της συγκεκριμένης εργασίας είναι οι εξής:

- Αρχικά, προτείνει τη χρήση γνωσιακών πρακτόρων λογισμικού για την εκπροσώπηση οντοτήτων του IoE ακολουθώντας την περίπτωση χρήσης του SIoT.
- Με βάση την πολυπρακτορική προσέγγιση παρουσιάζει την προτεινόμενη αρχιτεκτονική της αποκεντρωμένης πλατφόρμας για το SIoT.
- Καθορίζει την πολυεπίπεδη δομή του γνωσιακού πράκτορα του SIoT.
- Προτείνει τη χρήση τεχνικών Βαθιάς Μάθησης για την εξαγωγή ορθών πληροφοριών πλαισίου από ακατέργαστα δεδομένα αισθητήρων.
- Παρουσιάζει δύο οντολογίες για τη γνωσιακή διαχείριση των κοινωνικών σχέσεων μεταξύ των έξυπνων οντοτήτων και την επίτευξη των στόχων τους, καθώς και την κεντρική οντολογία για την αναπαράσταση των οντοτήτων του SIoT.
- Εισάγει μία διεπαφή χρήστη για τη δυναμική εισαγωγή σημασιολογικών κανόνων στο SIoT.
- Περιγράφει έναν αποκεντροποιημένο μηχανισμό εντοπισμού αντικρουόμενων κανόνων, βασισμένο σε μία οντολογία αναπαράστασης κανόνων και στον παραγωγικό συλλογισμό.
- Προτείνει, με βάση τις σχεδιαζόμενες οντολογίες, την ύπαρξη ενός ενσωματωμένου *αρχείου δήλωσης* σε έξυπνα αντικείμενα, το οποίο καθορίζει το προφίλ του έξυπνου αντικειμένου και ως εκ τούτου τον κύκλο ζωής του.
- Επιπλέον, στα πλαίσια της διατριβής υλοποιήθηκε ένα εργαλείο προσομοίωσης για το SIoT που ονομάζεται ASSIST (Agent Simulator for Social Smart Things) [19], το οποίο:

1. Απεικονίζει γραφικά τις κοινωνικές σχέσεις μιας οντότητας του SIoT.
  2. Δίνει τη δυνατότητα στους προγραμματιστές να πειραματισθούν σε διαφορετικές κοινωνικές συμπεριφορές των πρακτόρων λογισμικού τροποποιώντας τους κανόνες για τη διαχείριση φιλίας.
  3. Επιτρέπει στους προγραμματιστές να αλλάξουν τους κανόνες σχετικά με την ανακάλυψη υπηρεσιών και την πλοήγηση δικτύου.
- Ακόμα, αναπτύχθηκε ένα καινοτόμο Βαθύ Συνελικτικό Νευρωνικό Δίκτυο, το οποίο ονομάζεται PerceptionNet, με στόχο την αναγνώριση ανθρώπινων δραστηριοτήτων από ακατέργαστα δεδομένα κίνησης. Οι καινοτομίες του PerceptionNet είναι οι ακόλουθες:
    1. Αποδεικνύει ότι η καθυστερημένη συγχώνευση των δεδομένων των αισθητήρων είναι πιο αποτελεσματική από την πρώιμη, στα Βαθιά Συνελικτικά Νευρωνικά Δίκτυα.
    2. Εφαρμόζει μία συνάρτηση δισδιάστατης συνέλιξης σε σήματα κίνησης, τα οποία είναι στοιβαγμένα κατακόρυφα.
    3. Χρησιμοποιεί τη συνάρτηση ολικής υποδειγματοληψίας μέσης τιμής, αντί ενός πλήρως συνδεδεμένου στρώματος νευρώνων, ως προτελευταίο επίπεδο του δικτύου.
    4. Υπερβαίνει τις αποδόσεις των υπάρχουσών τεχνολογιών αιχμής στην αναγνώριση ανθρώπινων δραστηριοτήτων.
  - Τέλος, για τη δοκιμή του προτεινόμενου γνωσιακού μηχανισμού:
    1. Εξετάζει δύο περιπτώσεις χρήσης του SIoT, όπου οι κοινωνικοί δεσμοί σχηματίζονται χρησιμοποιώντας ένα σύνολο σημασιολογικών κανόνων.
    2. Εφαρμόζει το εκπαιδευμένο Βαθύ Συνελικτικό Νευρωνικό Δίκτυο σε δύο σύνολα δεδομένων κίνησης, επιτυγχάνοντας υψηλότερα ποσοστά ορθότητας από αυτά που αναφέρονται σε προηγούμενες έρευνες.
    3. Εξετάζει μία περίπτωση χρήσης του SIoT με στόχο τον εντοπισμό αντικρουόμενων σημασιολογικών κανόνων.
    4. Καταγράφει τις καθυστερήσεις του προτεινόμενου αποκεντριοποιημένου και ενός κεντριοποιημένου μηχανισμού παραγωγικού συλλογισμού, ανάλογα με το πλήθος των σημασιολογικών κανόνων ανά συσκευή.

### 1.3 Διάρθρωση της διατριβής

Η παρούσα διατριβή αποτελείται από έξι κεφάλαια, εκτός του παρόντος κεφαλαίου, το οποίο παρέχει τις εισαγωγικές πληροφορίες περί του θέματος και περιγράφει τα κίνητρα και την προσέγγιση της συγκεκριμένης διατριβής. Τα υπόλοιπα κεφάλαια έχουν ως εξής:

Στο δεύτερο κεφάλαιο, παρουσιάζεται το θεωρητικό υπόβαθρο. Συγκεκριμένα, περιγράφονται οι περιπτώσεις χρήσης του IoT, όπως είναι το Κοινωνικό Διαδίκτυο των Πραγμάτων και το Γνωσιακό Διαδίκτυο των Πραγμάτων, τα οποία υπήρξαν τα κίνητρα για την παρούσα διατριβή. Επιπλέον, παρουσιάζονται οι τεχνολογίες αιχμής για την επίτευξη της κοινωνικοποίησης των έξυπνων οντοτήτων, δηλαδή οι πράκτορες λογισμικού και οι τεχνικές ασύγχρονης διασύνδεσης των έξυπνων αντικειμένων, μέσω του προτύπου ανταλλαγής μηνυμάτων Δημοσίευσης-Συνδρομής. Στη συνέχεια, γίνεται λόγος για τις επιλεγμένες τεχνολογίες που απαρτίζουν το γνωσιακό μηχανισμό, έχοντας ως στόχο την επίγνωση πλαισίου. Ειδικότερα, περιγράφονται οι τεχνολογίες του Σημασιολογικού Ιστού, οι οποίες συμβάλλουν στην αναπαράσταση της πληροφορίας και κάνουν χρήση παραγωγικού συλλογισμού. Τέλος, παρουσιάζονται οι τεχνικές Βαθιάς Μάθησης που έχουν χρησιμοποιηθεί σε βιβλιογραφικές αναφορές, καθώς και οι συναρτήσεις που απαρτίζουν το μοντέλο που αναπτύχθηκε για την αναγνώριση ανθρώπινων δραστηριοτήτων.

Στο τρίτο κεφάλαιο, περιγράφεται η αρχιτεκτονική της προτεινόμενης πλατφόρμας για τη γνωσιακή διαχείριση των σχέσεων μεταξύ των οντοτήτων, η οποία αποτελεί και το αντικείμενο της διατριβής. Αρχικά, αναλύονται οι κύριες οντότητες της πλατφόρμας, (δηλ., οι γνωσιακοί πράκτορες λογισμικού), και αναλύονται λεπτομερώς τα επιμέρους συστατικά στοιχεία λογισμικού. Στη συνέχεια παρουσιάζονται τα οντολογικά σχήματα που σχεδιάστηκαν με σκοπό τη διαχείριση των κοινωνικών σχέσεων των έξυπνων οντοτήτων του SIoT, την επίτευξη των στόχων τους, και τη δυναμική εισαγωγή κανόνων συμπεριφοράς. Το κεφάλαιο κλείνει, εισάγοντας τον ορισμό του κύκλου ζωής ενός αντικειμένου και καταγράφοντας τα στάδια που τον απαρτίζουν.

Στο τέταρτο κεφάλαιο, παρουσιάζονται τα εργαλεία λογισμικού που αναπτύχθηκαν για την υλοποίηση του οικοσυστήματος του SIoT. Για τον καθορισμό των στόχων των συσκευών μέσω κανόνων, αναπτύχθηκε η διεπαφή Sem-SHUI, της οποίας οι λειτουργίες και ο μηχανισμός εντοπισμού αντικρουόμενων πολιτικών εξηγούνται. Έπειτα, παρουσιάζεται ο προσομοιωτής ASSIST, ο οποίος αναπτύχθηκε

για «δώσει ζωή» στην προτεινόμενη προσέγγιση, δίνοντας τη δυνατότητα στους χρήστες του να εξετάσουν σενάρια χρήσης.

Στο πέμπτο κεφάλαιο, παρουσιάζεται η αρχιτεκτονική του αναπτυχθέντος μοντέλου Βαθιάς Μάθησης, και περιγράφεται λεπτομερώς η πειραματική διαδικασία για την αξιολόγησή του. Συγκεκριμένα, εξετάζεται η ορθότητα του μοντέλου κάνοντας χρήση δύο συνόλων δεδομένων, τα οποία απαρτίζονται από σήματα αισθητήρων κίνησης.

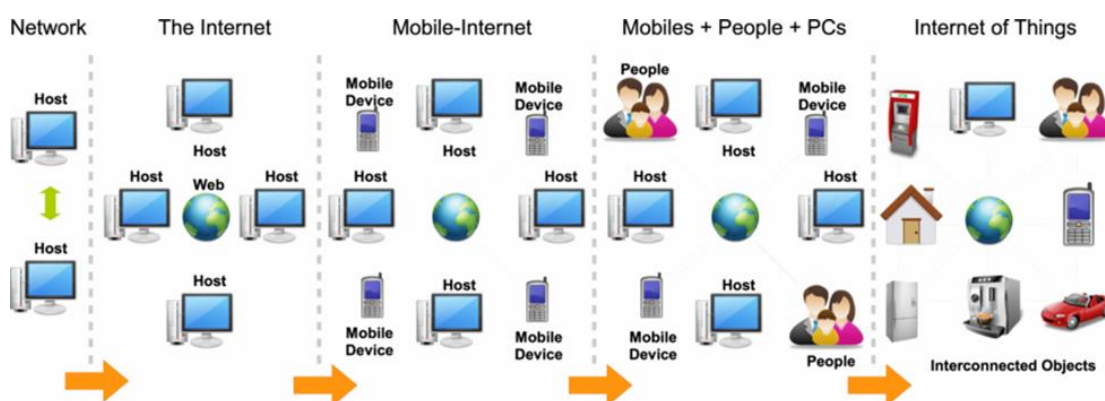
Στο έκτο κεφάλαιο, περιγράφεται ο τρόπος με τον οποίο η προαναφερθείσα πλατφόρμα για το SIoT μπορεί να εφαρμοσθεί σε οποιοδήποτε IoT περιβάλλον για τη δημιουργία δυναμικών σχέσεων φιλίας μεταξύ των οντοτήτων, δεδομένου ότι υπάρχουν/αξιοποιούνται δύο συνεργατικά σενάρια χρήσης με καλώς ορισμένους στόχους και ανάγκες, και κάνοντας χρήση του προσομοιωτή ASSIST. Θα πρέπει να σημειωθεί πως και στα δύο σενάρια οι πράκτορες λογισμικού του SIoT βασίστηκαν σε σημασιολογικούς κανόνες για να περιηγηθούν στο κοινωνικό δίκτυο και να κοινωνικοποιηθούν, προκειμένου να καλύψουν τις ανάγκες τους. Το κεφάλαιο κλείνει παρουσιάζοντας ένα σενάριο έξυπνου σπιτιού, με σκοπό τον εντοπισμό αντικρουόμενων κανόνων και καταγράφει τους χρόνους απόκρισης του προτεινόμενου μηχανισμού παραγωγικού συλλογισμού.

Τέλος, το έβδομο κεφάλαιο αποτελεί τον επίλογο της διατριβής. Στο κεφάλαιο αυτό, διατυπώνονται τα βασικά συμπεράσματα που εξήχθησαν στα πλαίσια της παρούσας διδακτορικής διατριβής, ενώ στη συνέχεια συνοψίζονται τα κύρια μελλοντικά σχέδια προς τη συνέχιση της διατριβής, της αξιοποίησης και του εμπλουτισμού τη λύσης που παρουσιάσθηκε στη διατριβή.

## 2 Θεωρητικό Υπόβαθρο

### 2.1 Διαδίκτυο των Πραγμάτων

Πριν διερευνηθεί σε βάθος το IoT, αξίζει να εξεταστεί η εξέλιξη του Διαδικτύου. Στα τέλη της δεκαετίας του 1960, η επικοινωνία μεταξύ δύο υπολογιστών κατέστη δυνατή μέσω ενός δικτύου υπολογιστών [20]. Στις αρχές της δεκαετίας του 1980 εισήχθη το Πρωτόκολλο Ελέγχου Μετάδοσης/Πρωτόκολλο Διαδικτύου (Transmission Control Protocol/Internet Protocol; TCP/IP), επιτρέποντας την εμπορική χρήση του Διαδικτύου στα τέλη της δεκαετίας του 1980. Αργότερα, ο Παγκόσμιος Ιστός (World Wide Web; WWW) έγινε διαθέσιμος το 1991, γεγονός που έκανε το Διαδίκτυο πιο δημοφιλές και τόνωσε την ταχεία ανάπτυξή του. Έπειτα, οι κινητές συσκευές άρχισαν να συνδέονται με το Διαδίκτυο και σχημάτισαν το Κινητό-Διαδίκτυο. Με την εμφάνιση της κοινωνικής δικτύωσης, οι χρήστες άρχισαν να συνδέονται μέσω του Διαδικτύου. Το τελευταίο βήμα ήταν το Διαδίκτυο των Πραγμάτων, όπου τα αντικείμενα γύρω μας μπορούν να συνδεθούν μεταξύ τους (π.χ., μηχανή με μηχανή) και να επικοινωνούν μέσω του Διαδικτύου. Η Εικόνα 5 απεικονίζει τις πέντε φάσεις της εξέλιξης του Διαδικτύου [21].



Εικόνα 5: Η εξέλιξη του Διαδικτύου σε πέντε φάσεις. Ξεκινά με τη σύνδεση δύο υπολογιστών και στη συνέχεια κινείται προς τη δημιουργία του Παγκόσμιου Ιστού με τη σύνδεση μεγάλου αριθμού υπολογιστών. Το Κινητό Διαδίκτυο προέκυψε συνδέοντας τις κινητές συσκευές στο Διαδίκτυο. Στη συνέχεια, οι άνθρωποι εντάχθηκαν στο Διαδίκτυο μέσω Κοινωνικών Δικτύων. Τέλος, κινείται προς το Διαδίκτυο των Πραγμάτων συνδέοντας καθημερινά αντικείμενα με το Διαδίκτυο [21].

Ο όρος "Διαδίκτυο των Πραγμάτων" εισήχθη για πρώτη φορά από τον Kevin Ashton [22], το 1998. Αναφέρει ότι το Διαδίκτυο των Πραγμάτων έχει τη δυνατότητα να αλλάξει τον κόσμο, όπως ακριβώς έκανε το Διαδίκτυο, και ίσως ακόμη περισσότερο. Αργότερα, το IoT εισήχθη επίσημα από τη Διεθνή Ένωση Τηλεπικοινωνιών (International Telecommunication Union; ITU) το 2005 [23]. Δυστυχώς, όμως, επειδή η έρευνα όσον αφορά το IoT είναι ακόμα σε πρωταρχικό

στάδιο, δεν υπάρχουν τυποποιημένοι ορισμοί για το IoT. Οι παρακάτω ορισμοί δόθηκαν από διάφορους ερευνητές:

- *"Τα πράγματα έχουν ταυτότητες και εικονικές προσωπικότητες που λειτουργούν σε έξυπνους χώρους χρησιμοποιώντας έξυπνες διεπαφές για σύνδεση και επικοινωνία μέσα στο κοινωνικό περιβάλλον και το πλαίσιο των χρηστών"* [24].
- *"Η σημασιολογική προέλευση της έκφρασης αποτελείται από δύο λέξεις και έννοιες: Διαδίκτυο των Πραγμάτων, όπου το Διαδίκτυο μπορεί να οριστεί ως το παγκόσμιο δίκτυο διασυνδεδεμένων δικτύων υπολογιστών, βασισμένο σε ένα πρότυπο πρωτόκολλο επικοινωνίας, (TCP/IP), ενώ το Πράγμα είναι ένα αντικείμενο που δεν προσδιορίζεται με ακρίβεια. Ως εκ τούτου, σημασιολογικά, το Διαδίκτυο των Πραγμάτων σημαίνει έναν παγκόσμιο δίκτυο διασυνδεδεμένων αντικειμένων με μοναδική διευθυνσιοδότηση, βασισμένο σε πρότυπα πρωτόκολλα επικοινωνίας* [2].
- *"Το Διαδίκτυο των Πραγμάτων επιτρέπει στους ανθρώπους και τα πράγματα να συνδέονται οποτεδήποτε, οπουδήποτε, με οτιδήποτε και οποιονδήποτε, ιδανικά χρησιμοποιώντας οποιαδήποτε διαδρομή/δίκτυο και οποιαδήποτε υπηρεσία"* [25].

Η συγκεκριμένη διατριβή βασίζεται στον τελευταίο ορισμό, ο οποίος στηρίζεται στην προσέγγιση τα Πάντα-ως-Υπηρεσία (Everything-as-a-Service) [26], η οποία είναι εξαιρετικά αποδοτική, κλιμακωτή και εύκολη στη χρήση. Πριν από δύο δεκαετίες, λοιπόν, ένας κόσμος όπου τα καθημερινά αντικείμενα μπορούσαν να αισθανθούν τον κόσμο μέσω αισθητήρων και στη συνέχεια να αναλύσουν, να αποθηκεύσουν και να ανταλλάξουν πληροφορίες ή υπηρεσίες, υπήρχε μόνο σε μυθιστορήματα επιστημονικής φαντασίας. Σήμερα, τέτοια σενάρια γίνονται όλο και περισσότερο πραγματικότητα, στα ενσωματωμένα συστήματα που καθιστούν τα αντικείμενα έξυπνα.

### **2.1.1 Περιπτώσεις χρήσης του Διαδικτύου των Πραγμάτων**

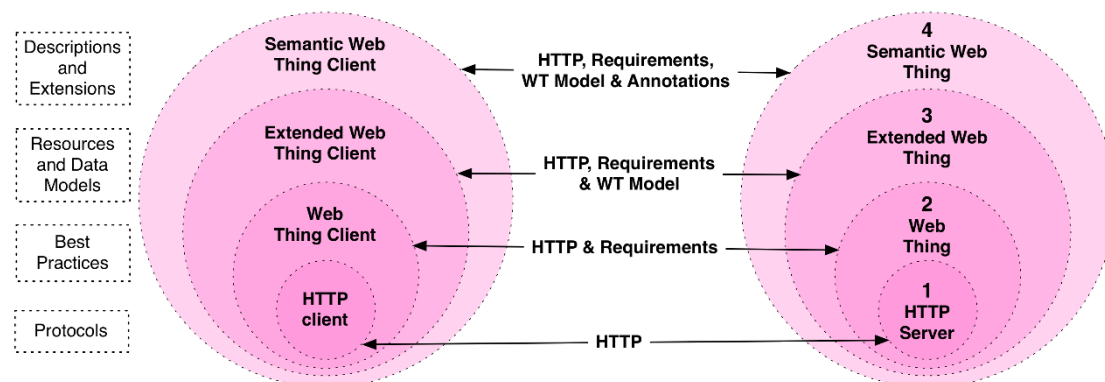
Οι ραγδαίες εξελίξεις στους τομείς των επικοινωνιών, της ηλεκτρονικής και της πληροφορικής, έχουν δημιουργήσει την ανάγκη για την υποστήριξη ενός οικοσυστήματος IoT που θα επιτρέπει τη δυναμική διασύνδεση των αντικειμένων, με στόχο την ανταλλαγή δεδομένων και υπηρεσιών. Υπάρχουν πολλοί όροι για να περιγράψουν αυτό το νέο οικοσύστημα και τις τεχνολογίες που το ενεργοποιούν. Αναφορικά η παρούσα διατριβή βασίζεται στις παρακάτω τεχνολογίες αιχμής για το IoT:

- Παγκόσμιος Ιστός των Πραγμάτων (WoT)
- Κοινωνικό Διαδίκτυο των Πραγμάτων (SIoT)

- Γνωσιακό Διαδίκτυο των Πραγμάτων (CIoT)

### 2.1.1.1 Παγκόσμιος Ιστός των Πραγμάτων

Ο Παγκόσμιος Ιστός των Πραγμάτων (Web of Things; WoT) είναι μία εξειδίκευση του IoT που χρησιμοποιεί αυτό που έκανε τον Παγκόσμιο Ιστό τόσο επιτυχημένο και το εφαρμόζει στις ενσωματωμένες συσκευές, προκειμένου να καταστήσει τις πιο πρόσφατες εξελίξεις στο IoT διαθέσιμες σε όσο το δυνατόν περισσότερους προγραμματιστές [1]. Πιο συγκεκριμένα, στο WoT, όπως και στον Παγκόσμιο Ιστό, οποιαδήποτε συσκευή ή αντικείμενο με ενσωματωμένο πρόγραμμα επεξεργασίας κειμένου και τη βασική κατανόηση των προτύπων Ιστού (HTML και HTTP) μπορεί να συνδεθεί στον Παγκόσμιο Ιστό. Επίσης, επιτρέπει τη μετάβαση στο επόμενο επίπεδο (Εικόνα 6) και βοηθά στη δημιουργία αποτελεσματικών διαδραστικών και καινοτόμων εφαρμογών που συνδυάζουν το φυσικό και τον ψηφιακό κόσμο [27].

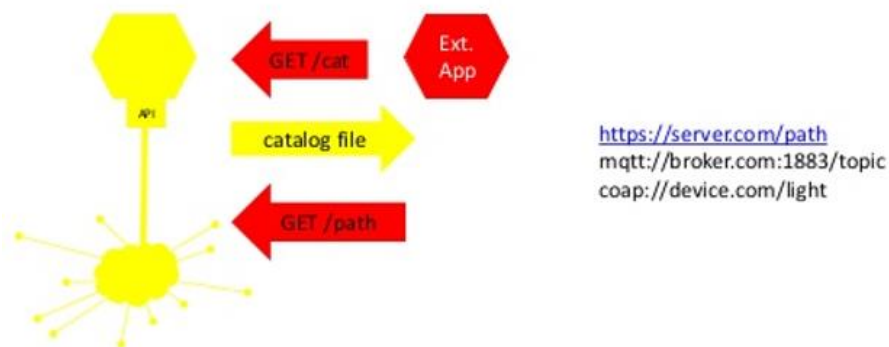


Εικόνα 6: Τα διάφορα επίπεδα περιγραφής του WoT [1].

Η Εικόνα 6 παρουσιάζει τα διάφορα επίπεδα του WoT. Στο 1<sup>ο</sup> επίπεδο βρίσκεται ο εξυπηρετητής (server) και ο πελάτης (client) HTTP. Στο 2<sup>ο</sup> επίπεδο βρίσκεται οποιαδήποτε συσκευή που μπορεί να έχει μία HTTP Διεπαφή Προγραμματισμού Εφαρμογών (Application Programming Interface; API). Στο 3<sup>ο</sup> επίπεδο βρίσκονται τα αντικείμενα που κάνουν χρήση ενός κοινόχρηστου μοντέλου αναπαράστασης δεδομένων, και στο 4<sup>ο</sup> επίπεδο γίνεται η προσθήκη σημασιολογικών σχολιασμών για να εξασφαλισθεί η διαλειτουργικότητα μεταξύ των έξυπνων αντικειμένων και να επιτραπεί στους HTTP clients να ανακαλύπτουν αυτόματα τις υπηρεσίες που τους ενδιαφέρουν.

Το τελευταίο επίπεδο του WoT καλείται και Σημασιολογικός Ιστός των Πραγμάτων (Semantic Web of Things; SWoT), καθώς συνδυάζει το WoT και τις τεχνολογίες του Σημασιολογικού Ιστού [28]. Συγκεκριμένα, η προσέγγιση αυτή μπορεί να επαναχρησιμοποιεί τα πρότυπα του Σημασιολογικού Ιστού για να περιγράψει το αντικείμενο (π.χ., το όνομα του αντικειμένου), τις ιδιότητες του

αντικειμένου (π.χ., την κατάσταση στην οποία βρίσκεται), τις εντολές που δέχεται, και τις υπηρεσίες που προσφέρει. Για παράδειγμα, περιγράφοντας τα δεδομένα τους με τη μορφή RDFa [29] ή JSON-LD [30] επιτρέπεται η αναζήτηση και των αντικειμένων μέσω μηχανών αναζήτησης για να επιτευχθεί η αυτοματοποιημένη αλληλεπίδραση τους (δηλ., η ανταλλαγή πληροφορίας). Ένα σημαντικό πρότυπο που βασίζεται στις τεχνολογίες του Σημασιολογικού Ιστού είναι το Hypercat [31], το οποίο προδιαγράφει λεπτομερώς τον τρόπο με τον οποίο θα πρέπει να περιγράφονται οι υπηρεσίες των έξυπνων αντικειμένων για να επιτευχθεί η εγγραφή τους σε καταλόγους υπηρεσιών και συνεπώς η αυτόματη πρόσβαση σε αυτούς, μέσω του Διαδικτύου, από εφαρμογές λογισμικού. Η Εικόνα 7 απεικονίζει την επικοινωνία μίας εφαρμογής με την RESTful API ενός έξυπνου αντικειμένου κάνοντας χρήση του πρωτοκόλλου HTTP. Στην αρχή η εφαρμογή ζητά από το έξυπνο αντικείμενο τον κατάλογο με τις υπηρεσίες του, και αφού τον λάβει, ζητά τη συγκεκριμένη υπηρεσία που την ενδιαφέρει.



Εικόνα 7: Απεικόνιση της πρόσβασης μίας εφαρμογής σε έναν κατάλογο υπηρεσιών IoT [31].

### 2.1.1.2 Κοινωνικό Διαδίκτυο των Πραγμάτων

Το Κοινωνικό Διαδίκτυο των Πραγμάτων είναι ένα κοινωνικό δίκτυο έξυπνων αντικειμένων. Παρόμοια με τον τρόπο με τον οποίο οι άνθρωποι συμμετέχουν στις Υπηρεσίες Κοινωνικών Δικτύων (Social Network Services; SNSs), μπορεί να υπάρξει ένα δίκτυο αποτελούμενο από τις κοινωνικές σχέσεις μεταξύ έξυπνων αντικειμένων. Πιο συγκεκριμένα, το SIoT πρότυπο προσφέρει: α) πλοήγηση στο δίκτυο για κλιμακούμενη και αποτελεσματική ανακάλυψη υπηρεσιών και αντικειμένων, β) εκμετάλλευση μοντέλων Κοινωνικών Δικτύων για την αντιμετώπιση ζητημάτων που σχετίζονται με το IoT και γ) δημιουργία ενός επιπέδου αξιοπιστίας μεταξύ των αντικειμένων, τα οποία είναι φίλοι. Η πρώτη έννοια είναι αυτό που κάνει το SIoT ένα τόσο ισχυρό πρότυπο. Κάθε αντικείμενο αναζητά στο δίκτυο μία επιθυμητή υπηρεσία, ρωτώντας τους φίλους του, τους φίλους των φίλων του και ούτω καθ'εξής, με έναν διανεμημένο τρόπο. Αυτή η ιδέα είναι που έκανε τα ανθρώπινα Κοινωνικά Δίκτυα τόσο χρήσιμα και πρωτοεμφανίστηκε από ένα πείραμα του Stanley Milgram, όπου σε μία προσπάθειά του να υπολογίσει το μέσο μήκος διαδρομής για τα



Κοινωνικά Δίκτυα των ανθρώπων συνειδητοποίησε ότι ο καθένας χωρίζεται από τον άλλο με έξι βαθμούς (συνδέσεις) το πολύ, οδηγώντας τον στη θεωρία των μικρών κόσμων [32].

Στην [33] η συγγραφική ομάδα (Atzori και λοιποί) προτείνει μία πιθανή αρχιτεκτονική για το SIoT αναλύοντας διάφορους τύπους πιθανών υπηρεσιών και εφαρμογών, με σκοπό την κατηγοριοποίηση των κοινωνικών σχέσεων ενός έξυπνου αντικειμένου στους εξής πέντε τύπους σχέσεων:

- Γονική Σχέση Αντικειμένων (Parental Object Relationship; POR): δημιουργείται μεταξύ αντικειμένων που ανήκουν στην ίδια παρτίδα παραγωγής, δηλαδή, συνήθως σε ομοιογενή αντικείμενα που κατασκευάστηκαν την ίδια χρονική περίοδο από τον ίδιο κατασκευαστή.
- Σχέση Συνεγκατάστασης Αντικειμένων (Co-location Object Relationship; C-LOR): δημιουργείται μεταξύ αντικειμένων (ομοιογενών ή ετερογενών) που χρησιμοποιούνται πάντα στον ίδιο τόπο (όπως στην περίπτωση των αισθητήρων και των ενεργοποιητών που χρησιμοποιούνται στο ίδιο περιβάλλον, όπως σε ένα έξυπνο σπίτι ή σε μία έξυπνη πόλη). Σε ορισμένες περιπτώσεις, οι σχέσεις τύπου C-LOR δημιουργούνται μεταξύ αντικειμένων που είναι απίθανο να συνεργαστούν για να επιτύχουν έναν κοινό στόχο. Παρόλα αυτά, είναι χρήσιμες ώστε να γεμίσει το δίκτυο με «σύντομες» συνδέσεις.
- Σχέση Συνεργασίας Αντικειμένων (Co-work Object Relationship; C-WOR): δημιουργείται μεταξύ αντικειμένων που συνεργάζονται για να παρέχουν μία κοινή εφαρμογή του IoT (όπως στην περίπτωση αντικειμένων που έρχονται σε επαφή για να χρησιμοποιηθούν μαζί για εφαρμογές όπως άμεση αντίδραση σε καταστάσεις έκτακτης ανάγκης, τηλεϊατρική, κλπ.).
- Σχέση Ιδιοκτησίας Αντικειμένων (Ownership Object Relationship; OOR): δημιουργείται μεταξύ ετερογενών αντικειμένων που ανήκουν στον ίδιο χρήστη (π.χ., κινητά τηλέφωνα, συσκευές αναπαραγωγής μουσικής, κονσόλες παιχνιδιών κ.λπ.).
- Κοινωνική Σχέση Αντικειμένων (Social Object Relationship; SOR): δημιουργείται μεταξύ αντικειμένων που έρχονται σε επαφή, σποραδικά ή συνεχώς, επειδή οι ιδιοκτήτες τους έρχονται σε επαφή μεταξύ τους κατά τη διάρκεια της ζωής τους (π.χ., συσκευές και αισθητήρες που ανήκουν σε φίλους, συμμαθητές, συντρόφους, συναδέλφους).

Από τους παραπάνω ορισμούς προκύπτει η ανάγκη ύπαρξης στρατηγικών για τη βέλτιστη επιλογή των σχέσεων φιλίας μεταξύ των αντικειμένων. Στην [34] γίνεται

μελέτη για πέντε πιθανές στρατηγικές για την επιλογή φίλιας με βάση τη δομή του δικτύου. Πρόκειται για μία καλά διερευνημένη προσέγγιση που οδηγεί στο συμπέρασμα ότι προτιμάται η ύπαρξη περισσότερων τυχαίων φίλων παρά η ύπαρξη μίας κλίκας φίλων. Στο οικοσύστημα του SIoT αυτό σημαίνει ότι οι περιστασιακοί φίλοι και μερικές φορές οι συνεργατικοί είναι πιο πολύτιμοι από τους άλλους τύπους φίλων. Ωστόσο, το παραπάνω συμπέρασμα οδηγεί σε ένα άλλο: οι φίλοι κυριότητας (δηλαδή οι έξυπνες οντότητες που ανήκουν στον ίδιο ιδιοκτήτη) είναι λιγότερο σημαντικοί, καθώς δημιουργούν μία κλίκα. Σε πολλές περιπτώσεις αυτό ισχύει, αλλά σε άλλες το αντικείμενο δεν θα είναι σε θέση να ικανοποιήσει τις εξατομικευμένες ανάγκες του ιδιοκτήτη του. Για παράδειγμα, εάν ένας πράκτορας υγείας έχει αφαιρέσει από τους φίλους του μία φορητή συσκευή που είναι υπεύθυνη για την παρακολούθηση των βημάτων του κατόχου του, τότε θα πρέπει να ρωτήσει τους φίλους του εάν είναι συνδεδεμένοι με κάποιον που μπορεί να δώσει απαντήσεις σχετικά με τον αριθμό των βημάτων του κατόχου του.

Μία πλατφόρμα δημιουργίας σημασιολογικών υπηρεσιών για το πρότυπο SIoT παρουσιάζεται στο [35]. Η προσέγγιση των συγγραφέων είναι παρόμοια με αυτή που παρουσιάζεται στη συγκριμένη διατριβή, ωστόσο η προσέγγισή τους βασίζεται σε RESTful API, ενώ η παρούσα διατριβή σε πράκτορες λογισμικού και στο μηχανισμό Δημοσίευσης-Συνδρομής. Οι συγγραφείς συμβάλλουν στο SIoT πρότυπο προτείνοντας τη σύγκλιση του Ιστού, των αντικειμένων και των χρηστών ακολουθώντας μία σημασιολογική προσέγγιση. Επιπλέον, θεωρούν τις Υπηρεσίες Ιστού (Web services) ως μέρος του SIoT, επιτρέποντας στις συσκευές να ανταλλάξουν δεδομένα και πληροφορίες, και να ενσωματωθούν στον Ιστό των Πραγμάτων [36]. Συνεπώς, θεωρούν πως τα έξυπνα αντικείμενα είναι πλήρως ενσωματωμένα στον Παγκόσμιο Ιστό χρησιμοποιώντας μικροσκοπικούς ενσωματωμένους διακομιστές Ιστού που μπορούν να παράγουν και να καταναλώνουν RESTful υπηρεσίες. Μία άλλη σημασιολογική προσέγγιση για το SIoT είναι αυτή του πλαισίου socialite [37]. Το πλαίσιο λογισμικού socialite αξιοποιεί την οντολογία *Relationship* και συγκεκριμένους σημασιολογικούς κανόνες, προκειμένου να επιτύχει μία πιο αποτελεσματική διαχείριση σχέσεων για το SIoT.

Τέλος, στην [38] παρουσιάζεται ένας γνωσιακός μηχανισμός λήψης αποφάσεων και εξατομικευμένων συστάσεων για το SIoT. Όπως και στην παρούσα διδακτορική διατριβή, η προσέγγιση της συγκεκριμένης συγγραφικής ομάδας βασίζεται στην οντολογική αναπαράσταση των πληροφοριών πλαισίου και προτείνει τη συμμετοχή όλων των οντοτήτων του IoE στο SIoT. Η αλληλεπίδραση μεταξύ του χρήστη και των αντικειμένων επιτυγχάνεται μέσω της εφαρμογής ThingsChat. Μία κεντρική οντότητα (που τρέχει σε Υπολογιστικό Σύννεφο), επεξεργάζεται τα κείμενα του χρήστη, συμπεραίνει την κατάσταση του χρήστη και στη συνέχεια, μέσω κανόνων αυτοματισμού, οι οποίοι ενεργοποιούνται ανάλογα με την περίσταση, επιτυγχάνει τη δημιουργία εξατομικευμένων συστάσεων για το άτομο. Αν και πρόκειται για μία πολύ

ενδιαφέρουσα ερευνητική εργασία, είναι απαραίτητη η συμμετοχή του χρήστη για την αναγνώριση της κατάστασής του, ενώ η λήψη των αποφάσεων γίνεται από μία κεντρική μονάδα επεξεργασίας, κάνοντας το προτεινόμενο σύστημα μη επεκτάσιμο.

### **2.1.1.3 Γνωσιακό Διαδίκτυο των Πραγμάτων**

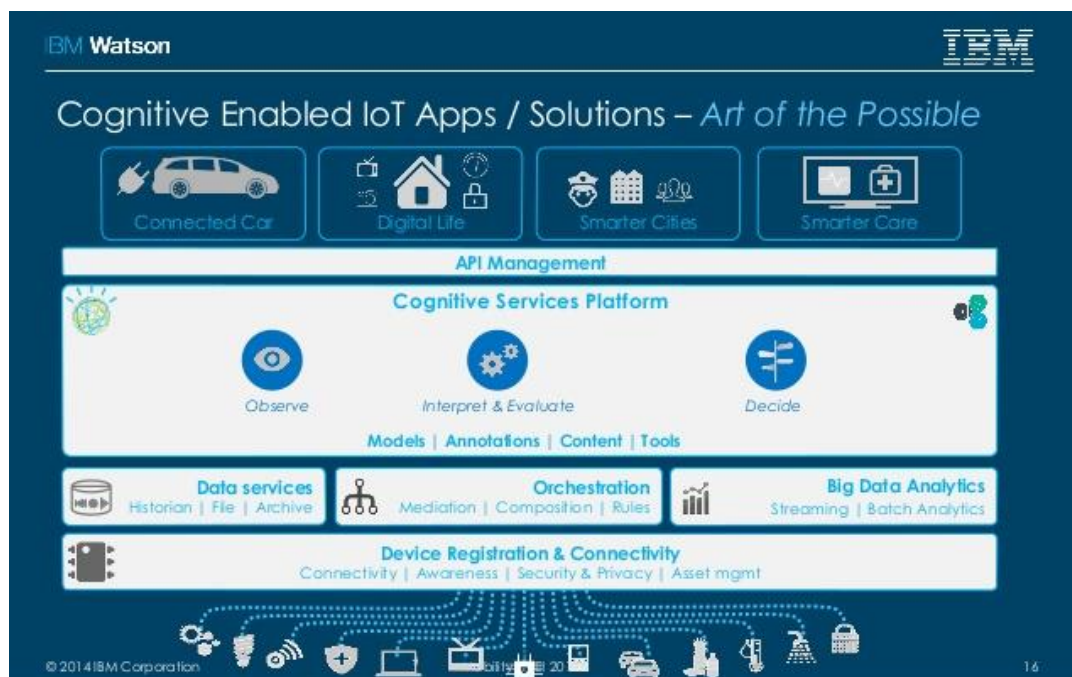
Ως γνωσιακή λειτουργία, ορίζεται η εφαρμογή της αποκτηθείσας γνώσης στις λειτουργίες των συσκευών, κατά κάποιο τρόπο όπως λειτουργεί ο ανθρώπινος εγκέφαλος για τη λήψη αποφάσεων. Για να αποκτηθούν οι γνώσεις πρέπει να συλλεχθούν δεδομένα και να εφαρμοσθούν μέθοδοι Τεχνητής Νοημοσύνης σε αυτά. Πιο συγκεκριμένα, η μετατροπή των συλλεχθέντων ακατέργαστων δεδομένων σε γνώση (πληροφορίες υψηλού επιπέδου) μπορεί να γίνει με το συλλογισμό, τη μοντελοποίηση και την εξαγωγή πληροφοριών πλαισίου μέσω τεχνικών Μηχανικής Μάθησης. Οι πληροφορίες πλαισίου μπορεί να χρησιμοποιηθούν για τη συγχώνευση (fusion) δεδομένων αισθητήρων από κοινού με στόχο την εξαγωγή νέων γνώσεων. Μόλις αποκτηθεί η γνώση, αυτή μπορεί να εφαρμοστεί για μια πιο έξυπνη αλληλεπίδραση και επικοινωνία.

Η έννοια του Γνωσιακού Διαδικτύου των Πραγμάτων (Cognitive Internet of Things; CIoT) εισάγεται στο [39] ως: “ένα IoT με γνωσιακούς και συνεργατικούς μηχανισμούς οι οποίοι ενσωματώνονται για τη βελτιστοποίηση της απόδοσης και την επίτευξη της νοημοσύνης”. Οι συγγραφείς στη [40] προτείνουν μία αρχιτεκτονική για το CIoT, λαμβάνοντας υπόψη την αναγκαιότητα ενός Κοινωνικού Δικτύου. Πιο συγκεκριμένα, προτείνουν μία αρχιτεκτονική τριών επιπέδων που αποτελείται από το φυσικό κόσμο, τον κυβερνοχώρο και τον κοινωνικό κόσμο. Ο κυβερνοχώρος αποτελείται από τέσσερα υποστρώματα:

1. το στρώμα της ανίχνευσης, στο οποίο συλλέγονται τα δεδομένα από το περιβάλλον,
2. το στρώμα της σημασιολογικής αναπαράστασης των δεδομένων, στο οποίο τα δεδομένα ανίχνευσης μεταφράζονται σε σημασιολογικά δεδομένα,
3. το στρώμα της λήψης αποφάσεων, όπου τα σημασιολογικά ορισμένα δεδομένα μεταβιβάζονται στους έξυπνους πράκτορες λογισμικού για λήψη αποφάσεων, και το
4. το στρώμα της αξιολόγησης της υπηρεσίας, στο οποίο οι παρεχόμενες υπηρεσίες αξιολογούνται από τους ανθρώπους.

Υποστηρικτής του εν λόγω παραδείγματος για το IoT είναι και η εταιρεία IBM, η οποία έχει αναπτύξει τον πράκτορα λογισμικού Watson [41], τον οποίο προσφέρει ως υπηρεσία μέσω της πλατφόρμας της IBM Bluemix [42]. Σύμφωνα με την IBM (Εικόνα 8), για την ανάπτυξη γνωσιακών εφαρμογών για το IoT απαιτούνται 4 στάδια επεξεργασίας των δεδομένων:

1. Παρατήρηση των φαινομένων, τα οποία διαδραματίζονται στο περιβάλλον του.
2. Ερμηνεία των φαινομένων και παραγωγή υποθέσεων για αυτά.
3. Αξιολόγηση των παραχθέντων ερμηνειών, κάνοντας χρήση τεχνικών μηχανικής μάθησης.
4. Λήψη αποφάσεων.

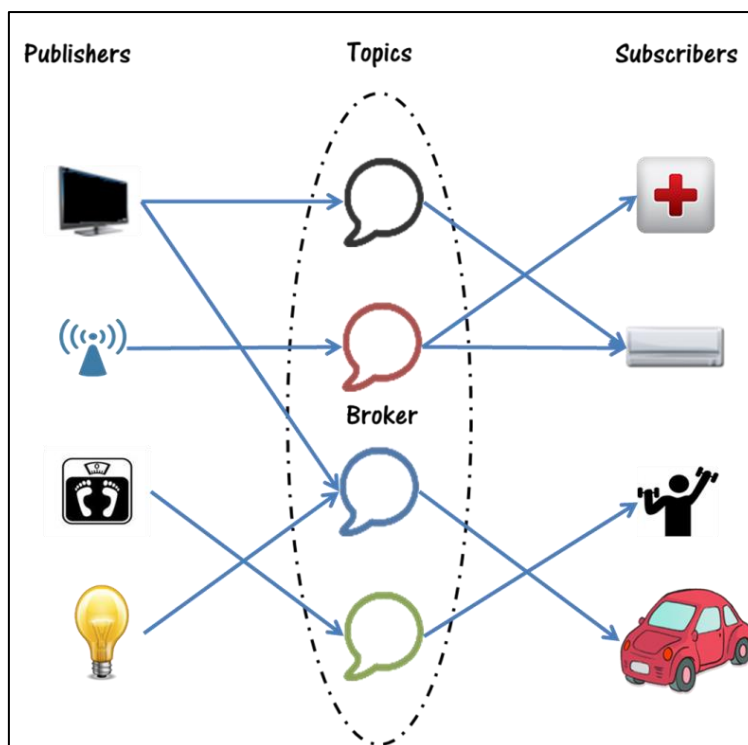


Εικόνα 8: Αρχιτεκτονική της IBM για το CIoT [43].

## 2.2 Υποστηρικτικές τεχνολογίες αιχμής για τη διασύνδεση των έξυπνων αντικειμένων

### 2.2.1 Πρότυπο ανταλλαγής μηνυμάτων Δημοσίευσης-Συνδρομής

Οι αρχιτεκτονικές που βασίζονται σε REST (Representational state transfer) υπηρεσίες ιστού, αν και αποτελούν μία από τις πιο διαδεδομένες προσεγγίσεις για την ανάπτυξη υπηρεσιοστρεφών εφαρμογών, χαρακτηρίζονται από ένα σημαντικό μειονέκτημα. Οι REST υπηρεσίες βασίζονται στο πρωτόκολλο HTTP, το οποίο υποστηρίζει σύγχρονη επικοινωνία αίτησης/απόκρισης. Αυτό συνεπάγεται πως δεν υποστηρίζουν απευθείας κοινοποιήσεις από τους εξυπηρετητές στους χρήστες. Οι συσκευές σε ένα IoT περιβάλλον, όπως και οι εφαρμογές με επίγνωση πλαισίου, έχουν ανάγκη για ασύγχρονη επικοινωνία μεταξύ τους και με τις πηγές δεδομένων, δηλαδή θα πρέπει να ακολουθούν αρχιτεκτονικές ανταλλαγής μηνυμάτων με ανάλογο ασύγχρονο τρόπο.



Εικόνα 9: Απεικόνιση του μηχανισμού Pub-Sub σε IoT περιβάλλον [44].

Το πρότυπο ανταλλαγής μηνυμάτων Δημοσίευσης-Συδρομής (Publish-Subscribe; Pub-Sub) [45] αποτελεί ένα μοντέλο όπου εφαρμογές και υπηρεσίες συνδέονται σε έναν κοινό τόπο, ο οποίος ονομάζεται Θέμα (Topic). Όλες οι οντότητες (π.χ., εφαρμογές) που «καταναλώνουν» το κοινό θέμα ονομάζονται Συδρομητές (Subscribers). Από την άλλη, κάθε οντότητα (π.χ., ένας αισθητήρας) μπορεί να αποστείλει ένα μήνυμα σχετικό με το θέμα, αναλαμβάνοντας έτσι το ρόλο του Εκδότη (Publisher). Το μήνυμα αποστέλλεται αυτόματα σε όλες τις εγγεγραμμένες οντότητες του θέματος με πλήρως ασύγχρονο τρόπο. Κάθε Συδρομητής μπορεί να είναι ταυτόχρονα και Εκδότης, δημιουργώντας έτσι ένα περιβάλλον ανταλλαγής μηνυμάτων, το οποίο είναι πλήρως παραμετροποιήσιμο και επεκτάσιμο. Για την επιτυχή ανταλλαγή και δρομολόγηση μηνυμάτων κάθε θέμα διατηρεί ένα Διαμεσολαβητή (Broker) ο οποίος κατέχει μία λίστα των συδρομητών, και αναλαμβάνει τη διαχείριση του θέματος, εξασφαλίζοντας ότι κάθε Συδρομητής θα πληροί συγκεκριμένα ποιοτικά χαρακτηριστικά που θα καθορίζουν τη παρουσία του και θα εξασφαλίζουν την ομαλή ανταλλαγή των μηνυμάτων. Η Εικόνα 9 παρουσιάζει το πρότυπο Pub-Sub σε ένα IoT περιβάλλον.

Το πρωτόκολλο επικοινωνίας MQTT (Message Queuing Telemetry Transport) [46] αποτελεί παράδειγμα αρχιτεκτονικής ανταλλαγής μηνυμάτων Pub-Sub, και είναι προσαρμοσμένο στην υποστήριξη δικτύων έξυπνων συσκευών (IoT), όπου μπορεί να παρατηρηθούν φαινόμενα υψηλής καθυστέρησης δικτύου (high latency).

## 2.2.2 Τεχνολογίες Σημασιολογικού Ιστού

Τα τελευταία χρόνια η εξέλιξη του Παγκόσμιου Ιστού, αλλά και του Διαδικτύου των Πραγμάτων γεννούν όλο και περισσότερες πηγές δεδομένων, κάτι που οδηγεί στην ανάγκη καθιέρωσης μίας προηγμένης προσέγγισης διαχείρισης της πληροφορίας. Ταυτόχρονα, κρίνεται αναγκαία η ενσωμάτωση του συνόλου των δεδομένων, καθώς με τον τρόπο αυτό είναι δυνατή η σφαιρική κατανόηση της καταναμημένης πληροφορίας, η συγχώνευση των πληροφοριών (information fusion) και οι σχέσεις που επικρατούν μεταξύ των ποικίλων τμημάτων της. Μία τεχνολογία, η οποία διασφαλίζει την αποδοτική διαχείριση και ενσωμάτωση ετερογενών δεδομένων και, κατ' επέκταση, την αποτελεσματική διασύνδεση εφαρμογών και συστημάτων, είναι η σημασιολογική περιγραφή των υποκείμενων δεδομένων και διαδικασιών. Αυτό συνεπάγεται την περιγραφή των προηγούμενων έτσι, ώστε το νόημά τους να είναι κατανοητό από μηχανές. Αυτή η θεμελιώδης ιδέα του σημασιολογικού εμπλουτισμού των δεδομένων οδήγησε στην εμφάνιση του Σημασιολογικού Ιστού (Semantic Web) [28]. Η ιδέα αυτή εισήχθη το 2001 από τον δημιουργό του Παγκόσμιου Ιστού (World Wide Web), Tim Berners-Lee.

Επιπλέον, το αρχικό όραμα της Κοινοπραξίας του Παγκοσμίου Ιστού (World Wide Web Consortium; W3C) ήταν, σύμφωνα με τον τρόπο που το εξέφρασε ο Tim Berners-Lee: *"Έχω ένα όνειρο για τον Ιστό [στον οποίο οι υπολογιστές] είναι σε θέση να αναλύσουν όλα τα δεδομένα στον Ιστό - το περιεχόμενο, τις σχέσεις, και τις συναλλαγές μεταξύ ανθρώπων και υπολογιστών. Ένας Σημασιολογικός Ιστός, ο οποίος καθιστά αυτό δυνατό, δεν έχει ακόμη προκύψει, αλλά όταν συμβεί αυτό, οι καθημερινοί μηχανισμοί του εμπορίου, της γραφειοκρατίας και της καθημερινής μας ζωής θα αντιμετωπιστούν από μηχανήματα που μιλούν με μηχανές. Οι έξυπνοι πράκτορες που οι άνθρωποι προβάλλουν εδώ και χρόνια θα υλοποιηθούν τελικά"*.

Έκτοτε, ο οργανισμός W3C ανέπτυξε ένα σύνολο προτύπων και εργαλείων με σκοπό να υποστηρίξει το όραμα αυτό. Το Πλαίσιο Περιγραφής Πόρων (Resource Description Framework; RDF) [47], το Σχήμα Πλαισίου Περιγραφής Πόρων (Resource Description Framework Schema – RDFS) [48], την Γλώσσα Οντολογίας Ιστού (Web Ontology Language; OWL) [49], καθώς και το πρωτόκολλο SPARQL (SPARQL Protocol and Query Language for RDF) [50] είναι τα βασικά εργαλεία που χρησιμοποιούνται για την αναπαράσταση/εξήγηση των δεδομένων ή των υπηρεσιών που προέρχονται από συσκευές.

Τα δεδομένα που ακολουθούν το RDF, καθώς και το RDFS, είναι εκφρασμένα ως τριπλέτες ή δηλώσεις (statements), οι οποίες έχουν τη μορφή *υποκείμενο - κατηγορήμα - αντικείμενο (object – predicates - subject)*. Κάθε δομικό στοιχείο μίας τριπλέτας αναγνωρίζεται μοναδικά από ένα Ενιαίο Αναγνωριστικό Πόρων (Uniform Resource Identifier; URI), γεγονός το οποίο διευκολύνει την ενσωμάτωση

κατανεμημένων δεδομένων αποφεύγοντας τον κίνδυνο συγκρούσεων. Θα πρέπει να σημειωθεί ότι τέτοιου είδους μορφές δεδομένων φιλοξενούνται σε αποθετήρια τριπλετών (triplestore). Πρόσβαση στα οντολογικά δεδομένων των αποθετηρίων τριπλετών δίνεται μέσω της SPARQL, της γλώσσας ερωτημάτων για οντολογικά δεδομένα. Η γλώσσα SPARQL φέρει αρκετές ομοιότητες με την turtle και την SQL (Structured Query Language), τη γλώσσα ερωτημάτων για σχεσιακά δεδομένα. Η κύρια λειτουργία ερωτημάτων εκτελείται μέσω της λέξης SELECT, επιστρέφοντας ένα σύνολο δηλώσεων οι οποίες ταιριάζουν με το μοτίβο που παρατέθηκε ως μέρος του ερωτήματος.

Η OWL αποτελεί μία γλώσσα αναπαράστασης γνώσης για το Σηματολογικό Ιστό. Λόγω των περιορισμών στη λειτουργικότητα του RDFS, ο οργανισμός W3C εισήγαγε την πρώτη έκδοση του προτύπου της OWL το 2004, ενώ το 2012 προτυποποιήθηκε η δεύτερη έκδοση της OWL (OWL 2) [51], η οποία είναι και η τρέχουσα. Η γλώσσα OWL δεν περιορίζεται απλά στην αναπαράσταση εννοιών και των συνδέσεών τους, αλλά στοχεύει στην ανάπτυξη οντολογιών, οι οποίες μπορεί να αξιοποιηθούν στα πλαίσια διαδικασιών λήψης αποφάσεων. Ταυτόχρονα, επωφελείται των δυνατοτήτων του RDF ώστε να επιτρέπει στις οντολογίες να περιέχουν αναφορές σε όρους άλλων οντολογιών OWL, δημιουργώντας τα Ανοικτά Συνδεδεμένα Δεδομένα (Linked Open Data; LOD). Ο Tim Berners-Lee όρισε την προϋποθέσεις που πρέπει να πληρούν τα δεδομένα, για να μπορούν να χαρακτηριστούν ως LOD, ως εξής [52]:

- Χρήση URIs για τη μοναδική αναγνώριση των πραγμάτων.
- Χρήση HTTP URIs ώστε να μπορούν να ανακαλυφθούν.
- Παροχή χρήσιμων πληροφοριών σχετικά με την ονομασία, μέσω της χρήσης ανοικτών προτύπων όπως των RDF, SPARQL, κτλ.
- Αναφορά σε άλλα URIs, ώστε να μπορούν να ανακαλυφθούν/ανακτηθούν περισσότερες πληροφορίες.

Συνεπώς, η OWL χρησιμοποιείται για την ανάπτυξη οντολογιών, δηλαδή για τον ορισμό εννοιών και των συσχετίσεων που υπάρχουν μεταξύ τους. Διαθέτει τη δυνατότητα δημιουργίας κλάσεων, ιδιοτήτων και στιγμιότυπων κλάσεων. Οι προαναφερθείσες ιδιότητες χωρίζονται σε ιδιότητες αντικειμένων (object property), οι οποίες συνδέουν στιγμιότυπα κλάσεων, και σε ιδιότητες δεδομένων (data property), οι οποίες αναθέτουν κάποια τιμή σε ένα στιγμιότυπο. Σε αντίθεση με το RDFS, το εύρος των δυνατοτήτων της γλώσσας OWL επεκτείνεται περαιτέρω, προσφέροντας λειτουργίες έκφρασης τομής και ένωσης κλάσεων και περιορισμών τιμών ή πληθικότητας. Συγκεκριμένα, η OWL 2 προσφέρεται για εφαρμογές που χρησιμοποιούν οντολογίες με μεγάλο αριθμό κλάσεων ή/και ιδιοτήτων. Οι γλώσσες οντολογιών ιστού και γενικά οι τεχνολογίες σηματολογικού ιστού χρησιμοποιούνται ευρέως και για την αναπαράσταση δεδομένων και οντοτήτων του IoT.

Η αρχική προσέγγιση των ερευνητών, καθώς δεν ήταν διαδεδομένη η έννοια του IoT, ήταν να προτείνουν οντολογίες για συγκεκριμένους τομείς όπως οι έξυπνες συσκευές και η διαχείριση έξυπνων κτιρίων, έχοντας ως κύριο στόχο την επίτευξη της διαλειτουργικότητας μεταξύ των διαφορετικών πρωτοκόλλων επικοινωνίας. Αξιοσημείωτη οντολογία πάνω στα έξυπνα αντικείμενα και τα πρωτόκολλα τους αποτελεί η Dog-Ont [53][54], καθώς και η «πρόδρομος» της DomoML-env [55]. Παρά το γεγονός ότι προτάθηκαν πολλές οντολογίες για την περιγραφή των έξυπνων αντικειμένων, μόνο μία κατάφερε να προτυποποιηθεί, καθώς είναι πολύ επεκτάσιμη.

Συγκριμένα, το 2012, η W3C πρότεινε μία πρότυπη οντολογία, την οντολογία SSN (Semantic Sensor Network) [56], για την περιγραφή των πηγών των αισθητήρων και των δεδομένων που συλλέχθηκαν μέσω αυτών ως παρατηρήσεις. Η οντολογία SSN στοχεύει στην επίλυση των προβλημάτων ετερογένειας που σχετίζονται με την ανακάλυψη αισθητήρων και τη συλλογή των δεδομένων, αλλά έχει περιορισμένες έννοιες για τη στήριξη της χωρικής και χρονικής συσχέτισης δεδομένων αισθητήρων με τους πόρους. Επίσης, είναι δύσκολο να περιγραφούν οι διαφορετικές δυνατότητες αισθητήρων χρησιμοποιώντας την SSN, καθώς υπάρχει μόνο μία κλάση για την περιγραφή του αισθητήρα. Συνεπώς, υπάρχει η ανάγκη μίας οντολογίας που θα περιγράφει με περισσότερη λεπτομέρεια τους αισθητήρες και τις υπόλοιπες συσκευές του IoT.

Ακολουθώντας το παράδειγμα της SSN, έχουν προταθεί πολλές επεκτάσιμες, οντολογίες, που δεν βασίζονται σε ένα συγκεκριμένο τομέα, για την αναπαράσταση των οντοτήτων του IoT. Ειδικότερα, αρκετές ερευνητικές δουλειές έχουν αρχίσει να επαναχρησιμοποιούν τις υπάρχουσες οντολογίες, συνδυάζοντάς τις για να προτείνουν νέες οντολογίες για το IoT. Αυτές οι νέες οντολογίες προορίζονται για ειδικές πλατφόρμες που στοχεύουν στη συλλογή και ενσωμάτωση δεδομένων IoT. Η οντολογία OpenIoT [57] χρησιμοποιεί την SSN ως βάση για να βασιστεί σε έννοιες που απαιτούνται για εφαρμογές IoT – Παρατήρηση (Observation), Αισθητήρας (Sensor) και Τοποθεσία (Location). Η οντολογία FIESTA-IoT [58] είναι μία ακόμη προσπάθεια να ενοποιηθούν οι υπάρχουσες οντολογίες (SSN και IoT-lite [59]) για τον τομέα του IoT. Ωστόσο, καμία από τις δύο δεν κάνει χρήση εννοιών για τη διαχείριση των κοινωνικών σχέσεων μεταξύ των αντικειμένων, όπως κάνει η οντολογία FOAF (Friend of a Friend) [60] στην περίπτωση των ανθρώπων, και δεν εκφράζουν τις ανάγκες/στόχους των οντοτήτων του IoT.

### 2.2.3 Συλλογισμός σε οντολογικά δεδομένα

Η αναπαράσταση της γνώσης είναι το πεδίο της Τεχνητής Νοημοσύνης που επικεντρώνεται στο σχεδιασμό αναπαραστάσεων για ηλεκτρονικούς υπολογιστές που συλλαμβάνουν πληροφορίες σχετικά με τον κόσμο και μπορούν να χρησιμοποιηθούν για την επίλυση σύνθετων προβλημάτων. Η αιτία για την



αναπαράσταση της γνώσης είναι ότι ο συμβατικός δομημένος/διαδικαστικός κώδικας δεν είναι ο καλύτερος φορμαλισμός που θα χρησιμοποιηθεί για την επίλυση σύνθετων προβλημάτων. Η αναπαράσταση γνώσης καθιστά ευκολότερο τον προσδιορισμό του σύνθετου λογισμικού, διευκολύνει τη διατήρησή του από το διαδικαστικό κώδικα και μπορεί να χρησιμοποιηθεί σε συστήματα εμπειρογνομόνων. Επιπλέον, συμβαδίζει με την αυτοματοποιημένη συλλογιστική γιατί ένας από τους κύριους σκοπούς της αναπαράστασης της γνώσης είναι να μπορεί να συλλογίζεται με βάση αυτή τη γνώση, να καταλήγει σε συμπεράσματα, να εισάγει νέες γνώσεις κλπ. Σχεδόν όλες οι γλώσσες εκπροσώπησης της γνώσης έχουν έναν μηχανισμό συλλογισμού ή συμπερασμού ως μέρος του συστήματος. Βάση για το συλλογισμό σε οντολογικά δεδομένα αποτελεί ο Προτασιακός Λογισμός (Propositional Logic).

Ο Προτασιακός Λογισμός είναι η λογική που μελετά τις ατομικές προτάσεις όπως  $A, B$ , οι οποίες έχουν πάντα τιμές *σωστό* ή *λάθος*, και συνδέονται μεταξύ τους μέσω λογικών συνδέσμων, όπως *όχι* (άρνηση) ( $\neg A$ ), *και* (σύζευξη) ( $A \wedge B$ ), *ή* (διάζευξη) ( $A \vee B$ ), και *αν* (συνεπαγωγή) ( $A \rightarrow B$ ). Ο Προτασιακός Λογισμός χαρακτηρίζεται από ορθότητα (αποκομίζει μόνο σωστά αποτελέσματα), και πληρότητα (μπορεί να παράγει οποιοδήποτε λογικά έγκυρη φόρμουλα). Επιπλέον, πρόκειται για μία συλλογιστική που κάνει χρήση:

- *κατηγορημάτων (predicates)*, όπως  $P(x, y)$ , το οποίο αναπαριστά σχέσεις, δηλαδή, παράγει *σωστό* ή *λάθος* για ένα συνδυασμό τιμών των όρων  $x$  και  $y$ ,
- *συμβόλων ποσόδειξης (quantifiers)*, όπως  $\forall, \exists$ , και
- *συμβόλων συναρτήσεων (functions)*, όπως  $f(x), g(y, z)$ .

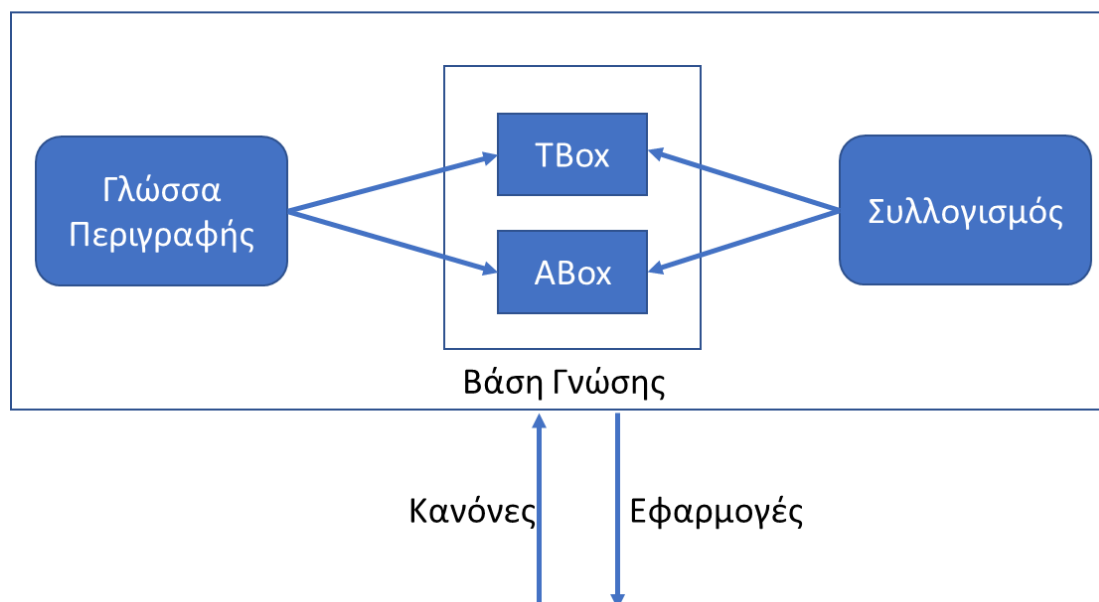
Οπότε, ο Προτασιακός Λογισμός μπορεί να σχηματίσει λογικές προτάσεις όπως  $\forall x \exists y (P(x) \rightarrow Q(f(y)))$ . Ο Προτασιακός Λογισμός πρώτου βαθμού είναι μία μορφή συλλογιστικής όπου οι ποσοτικοί δείκτες μπορούν να κυμαίνονται μόνο πάνω από ένα πεδίο τιμών. Σε λογικές υψηλότερης τάξης, οι ποσοτικοί δείκτες μπορούν να κυμαίνονται μεταξύ συνόλων τιμών, λειτουργιών και άλλων αντικειμένων. Έτσι, για παράδειγμα, η πρόταση ότι "κάθε σύνολο πραγματικών αριθμών έχει ένα ελάχιστο" δεν μπορεί να εκφραστεί σε λογική πρώτης τάξης, επειδή η ποσοτικοποίηση κυμαίνεται πάνω σε σύνολα, όχι σε καθορισμένα στοιχεία.

Η Περιγραφική Λογική (Description Logic) αναφέρεται σε ένα υποσύνολο της κατηγορηματικής λογικής πρώτης τάξης (first-order predication logic), δηλαδή σε μία οικογένεια φορμαλισμού εκπροσώπησης γνώσης που αντιπροσωπεύει τη γνώση ενός πεδίου εφαρμογής («κόσμος»), καθορίζοντας πρώτα τις σχετικές έννοιες του πεδίου (την ορολογία του), και στη συνέχεια χρησιμοποιώντας αυτές τις έννοιες για να καθορίσει τις ιδιότητες αντικειμένων και ατόμων που εμφανίζονται στο

συγκεκριμένο πεδίο (η περιγραφή του κόσμου). Με αυτή τη λογική, οι έννοιες, οι ρόλοι, τα άτομα και οι σχέσεις τους μπορούν να διαμορφωθούν. Η έννοια είναι μία συλλογή αντικειμένων και ο ρόλος είναι μία σχέση μεταξύ αντικειμένων. Όπως υποδηλώνει το όνομα της Περιγραφικής Λογικής (ΠΛ), ένα από τα χαρακτηριστικά αυτής της γλώσσας είναι ότι έχει σχεδιαστεί για να αντιπροσωπεύει κυρίως τη σημασιολογία.

Η γλώσσα OWL βασίζεται στην Περιγραφική Λογική, και χαρακτηριστικό της είναι η εφαρμογή κανόνων παραγωγικού συλλογισμού τόσο στα δεδομένα (ABox) όσο και στο σχήμα της οντολογίας (TBox). Η Εικόνα 10 παρουσιάζει ένα σύστημα Περιγραφικής Λογικής, όπου εισάγονται κανόνες σε μία βάση γνώσεων, ακολουθεί ο συλλογισμός με βάση τα TBox και ABox και τέλος, η εξαγόμενη γνώση χρησιμοποιείται για την ανάπτυξη έξυπνων εφαρμογών.

### Σύστημα Περιγραφικής Λογικής



Εικόνα 10: Εφαρμογή τεχνικών συλλογισμού στα σύνολα TBox και ABox.

Ως σώμα ορολογίας (terminological box, TBox) ή οντολογία (ontology) ορίζεται ένα σύνολο από αξιώματα υπαγωγής και ισοδυναμίας εννοιών και ρόλων, τα οποία ονομάζονται και αξιώματα ορολογίας. Για παράδειγμα, η κλάση της γυναίκας και του άντρα μπορούν να ορισθούν ως εξής:

Γυναίκα  $\equiv$  Άνθρωπος  $\wedge$  Θηλυκό

Άντρας  $\equiv$  Άνθρωπος  $\wedge$   $\neg$  Γυναίκα

Το σύνολο από ισχυρισμούς ισότητας ή ανισότητας στιγμιοτύπων, ισχυρισμούς εννοιών και ισχυρισμούς ρόλων ονομάζεται σώμα ισχυρισμών (assertion box, ABox). Το ABox περιέχει γνωσιακές γνώσεις (εξ ου και ο όρος "ABox"), που αφορούν ειδικά στα άτομα του τομέα του λόγου. Ένας τυπικός ισχυρισμός στο ABox είναι όταν κάποιος δηλώνει ότι ένα άτομο είναι μία περίπτωση μίας συγκεκριμένης έννοιας. Για παράδειγμα η σχέση: Γυναίκα('Μαρία') δηλώνει πως το άτομο 'Μαρία' είναι περίπτωση της έννοιας Γυναίκα.

Επίσης, η ΠΛ υποστηρίζει τη δημιουργία υποθέσεων όσον αφορά τα άτομα του κόσμου ο οποίος μοντελοποιείται, δηλαδή συντελεί στο να καθορισθούν σχέσεις στιγμιοτύπου (instance relations) ανάμεσα σε ένα άτομο (ζευγάρι ατόμων) και μία έννοια (ρόλο), οι οποίες ονομάζονται ισχυρισμοί (assertions). Για παράδειγμα η σχέση: έχειΠαιδί('Μαρία', 'Ιωάννα') δηλώνει πως το άτομο 'Μαρία' έχει παιδί το άτομο 'Ιωάννα'.

Η αναπαράσταση δεδομένων δεν αφορά αποκλειστικά στη μοντελοποίησή τους με τη χρήση οντολογιών. Εξέχοντα ρόλο διαδραματίζουν οι σημασιολογικοί κανόνες και ο παραγωγικός συλλογισμός (inference). Μέσω του παραγωγικού συλλογισμού, εξάγεται γνώση σχετική με γεγονότα τα οποία δεν έχουν δηλωθεί ρητά. Η OWL έχει σχεδιαστεί με τέτοιο τρόπο, ώστε να υποστηρίζει διάφορες μεθόδους παραγωγικού συλλογισμού, πάνω στα TBox και ABox.

Συλλογισμός σε TBox [61]:

- Υπαγωγή (subsumption). Το βασικό συμπέρασμα για τις εκφράσεις ιδεών στην Περιγραφική Λογική είναι η υπαγωγή (π.χ., η έννοια  $C$  υπάγεται στην έννοια  $D$ ), τυπικά γραμμένη ως  $C \subseteq D$ . Η υπαγωγή είναι ο έλεγχος του κατά πόσο η έννοια που δηλώνεται από το  $D$  (το υπερσύνολο) θεωρείται γενικότερη από την έννοια που υποδηλώνεται από το  $C$  (το υποσύνολο). Με άλλα λόγια, ο υποτομέας ελέγχει αν η πρώτη έννοια πάντα υποδηλώνει ένα υποσύνολο του συνόλου που υποδηλώνεται από το δεύτερο.
- Ικανοποιησιμότητα (satisfiability). Ένας άλλος τύπος συμπερασμάτων σχετικά με τις εκφράσεις της έννοιας είναι η ικανοποιησιμότητα, η οποία επικεντρώνεται στην επαλήθευση εάν μία έκφραση δεν ισοδυναμεί με το κενό σύνολο,  $C \neq \emptyset$ . Στην πραγματικότητα, η ικανοποιησιμότητα είναι μία ειδική περίπτωση υπαγωγής, με το υπερσύνολο να είναι το κενό σύνολο, που σημαίνει ότι μία έννοια δεν είναι ικανοποιήσιμη.
- Ισοδυναμία (equivalence). Δύο έννοιες είναι δυνατόν να ταυτίζονται, αν οι ιδιότητές τους καθορίζουν πάντα το ίδιο σύνολο ατόμων. Πιο συγκεκριμένα, δύο έννοιες  $C$  και  $D$  είναι ισοδύναμες σύμφωνα με ένα Tbox  $T$ , συμβολικά  $C \equiv D$ , αν για κάθε αντειμένο  $I$  του συνόλου  $T$ , ισχύει  $C^I \equiv D^I$ .

- Ανομοιότητα (disjointness). Δύο έννοιες είναι δυνατόν να χαρακτηρισθούν ως ανόμοιες, αν οι ιδιότητές τους καθορίζουν πάντα διαφορετικό σύνολο ατόμων. Πιο συγκεκριμένα, δύο έννοιες  $C$  και  $D$  είναι ανόμοιες σύμφωνα με ένα Tbox  $T$ , συμβολικά  $C \neq D$ , αν για κάθε αντειμένο  $I$  του συνόλου  $T$ , ισχύει  $C^I \neq D^I$ .

Συλλογισμός σε ABox [61]:

- Συνέπεια (consistency). Ένα Abox  $A$  είναι συνεπές σε σχέση με ένα Tbox  $T$  αν υπάρχει αντικείμενο του μοντέλου  $T$  που ικανοποιεί κάθε υπόθεση του  $A$ .
- Συνεπαγωγή (entailment). Ένα Abox  $A$  συνεπάγεται μία υπόθεση  $\varphi$ , συμβολικά  $A \models \varphi$ , αν κάθε διερμηνεία που ικανοποιεί το  $A$  ικανοποιεί και την υπόθεση  $\varphi$ .

Η υπαγωγή μεταξύ των εκφράσεων της έννοιας αναγνωρίστηκε ως το βασικό συμπέρασμα και η βασική αρχή των πρώτων αλγορίθμων υπαγωγής ήταν να μετασχηματίσει δύο έννοιες εισόδου σε γράφους με ετικέτες και να επαληθεύσει εάν θα μπορούσε κάποια να ενσωματωθεί στην άλλη· ο ενσωματωμένος γράφος θα αντιστοιχεί στη γενικότερη έννοια. Πρόκειται για προσέγγιση διαρθρωτικής σύγκρισης και η σχέση μεταξύ των εννοιών είναι μία δομική υπαγωγή.

Η βασική ιδέα είναι ότι τα στιγμιότυπα μίας σύνθετης έννοιας έχουν ένα σύνολο από ιδιότητες που απορρέουν από την έκφρασή της. Κάθε ιδιότητα αποτελεί και έναν περιορισμό, ο οποίος πρέπει να ικανοποιείται. Κάποιοι από τους περιορισμούς αυτούς μπορεί να έρχονται σε σύγκρουση (να δημιουργούν αντιφάσεις) και συνεπώς να μην είναι δυνατό να κατασκευασθεί ερμηνεία με κάποιο στιγμιότυπο της έννοιας. Στόχος της μεθόδου είναι να ελεγχθούν οι περιορισμοί που εμπλέκονται στην έκφραση της σύνθετης έννοιας, συγκεκριμένα να ελεγχθεί αν μπορούν να ικανοποιούνται ταυτόχρονα, ώστε η έννοια της οποίας η ικανοποιησιμότητα ελέγχεται να μπορεί να έχει ένα μη κενό σύνολο στιγμιότυπων σε κάποια ερμηνεία.

Τέλος, πρέπει να σημειωθεί ότι η διαδικασία της εξαγωγής γνώσης λαμβάνει χώρα μέσω του εργαλείου παραγωγικού συλλογισμού (reasoner). Ορισμένα εργαλεία παραγωγικού συλλογισμού είναι ενσωματωμένα σε εργαλεία μεγαλύτερου εύρους λειτουργιών, ενώ άλλα διατίθενται μεμονωμένα και μπορούν να λειτουργήσουν συνεργατικά με βιβλιοθήκες ή αποθετήρια τριπλετών. Ορισμένα από τα πιο διαδεδομένα εργαλεία διαχείρισης οντολογιών, τα οποία συνοδεύονται από εργαλεία παραγωγικού συλλογισμού, είναι το Apache Jena [62], το Sesame [63] και το Kaon2 [64], ενώ ορισμένα από τα πιο δημοφιλή αυτόνομα εργαλεία αυτής της κατηγορίας είναι το Pellet [65], το Fact++ [66], το RacerPro [67] και το Hermit [68].

## 2.2.4 Μηχανισμοί ανίχνευσης συγκρούσεων σε συστήματα αυτοματισμού

Ως αντιφατικοί ή συγκρουόμενοι κανόνες αυτοματισμού ορίζονται δύο κανόνες που έχουν ως συνέπεια την ταυτόχρονη επιβολή δύο ή και παραπάνω εντολών σε μία συσκευή, οι οποίοι δεν μπορούν να εξυπηρετηθούν ταυτόχρονα (π.χ., εντολές On και Off). Οι μέθοδοι ανίχνευσης συγκρούσεων στο πεδίο της αλληλεπίδρασης κανόνων, ταξινομούνται ως εκτός σύνδεσης (offline), κατά τη διάρκεια ορισμού ενός κανόνα, ή σε απευθείας σύνδεση (online), κατά τη διάρκεια ενεργοποίησης ενός κανόνα). Οι offline τεχνικές απαιτούν εξαντλητική γνώση του συστήματος (π.χ., να παρουσιάζονται σημασιολογικά όλα τα συστατικά στοιχεία), ενώ οι online τεχνικές ασχολούνται με τα συγκρουόμενα γεγονότα όταν εκτελούνται [69].

Οι συγγραφείς στην [70] προτείνουν ένα εργαλείο ελέγχου βασισμένο σε λογική πρώτης τάξης για την αυτοματοποιημένη ανίχνευση συγκρούσεων στα συστήματα αυτοματισμού οικιακής χρήσης. Ο Nakamura και λοιποί [71] επισημαίνουν ότι ένας online μηχανισμός επίλυσης συγκρούσεων βασίζεται σε τρεις έννοιες: (α) ενεργοποίηση, (β) υποχρεωτικές μέθοδοι και (γ) μηχανισμός αναστολής/επανάναρξης. Η πρώτη έννοια ορίζει το χρόνο έναρξης και λήξης εκτέλεσης των υπηρεσιών, η δεύτερη έννοια προσδιορίζει τις βασικές και τις προαιρετικές λειτουργίες στις υπηρεσίες, ενώ η τρίτη επιτρέπει στις υπηρεσίες χαμηλότερης προτεραιότητας να «κοιμούνται» προσωρινά και να «ξυπνούν» εάν τελειώσουν όλες οι υπηρεσίες που συγκρούονται με αυτές.

Ένα σύστημα οικιακού αυτοματισμού, το Homer, εισάγεται στην [72] μαζί με τη γλώσσα έκφρασης πολιτικών Homeric, από τους Maternaghan και Turner. Οι ίδιοι συγγραφείς [69], κάνοντας χρήση της Homeric, ακολουθούν μία προσέγγιση ανίχνευσης διενέξεων εκτός σύνδεσης. Το σύστημά τους ασχολείται με τις συγκρούσεις καθώς δημιουργούνται κανόνες, οδηγώντας στην εκτέλεση μόνο μη-αντιφατικών πολιτικών.

Ένα σύστημα οικιακού αυτοματισμού βασισμένο σε οντολογίες προτείνεται στην [73]. Το εν λόγω σύστημα αποτελείται από τέσσερα συστατικά: α) συγκέντρωση πληροφοριών πλαισίου, β) ανίχνευση συγκρούσεων, γ) επίλυση συγκρούσεων και δ) λήψη αποφάσεων. Το σύστημα χρησιμοποιεί μία οντολογία ως βάση γνώσεων και τα ερωτήματα SPARQL εξάγει πληροφορίες σχετικά με πιθανά συμβάντα σύγκρουσης. Αυτές οι πληροφορίες είναι η είσοδος της ενότητας επίλυσης συγκρούσεων, η οποία θεωρεί τις συγκρούσεις ως Προβλήματα Ικανοποίησης Περιορισμών (Constraint Satisfaction Problems; CSPs). Τέλος, η μονάδα λήψης αποφάσεων λαμβάνει τη λύση από τη μονάδα επίλυσης συγκρούσεων και τη μεταφράζει σε ενέργειες των ενεργοποιητών.

Τέλος, στην [74] παρουσιάζεται ένας μηχανισμός εντοπισμού offline συγκρούσεων που έχει ως στόχο να εντοπίσει αν υπάρχει μία κατάσταση που θα μπορούσε να ενεργοποιήσει πολλαπλούς κανόνες, οι οποίοι θα έχουν ως συνέπεια την επιβολή αντικρουόμενων εντολών. Ο εν λόγω μηχανισμός βασίζεται στη στρατηγική ανάλυσης άσπρου κουτιού (white box testing). Πιο συγκεκριμένα, κάνοντας χρήση του εργαλείου Rex [75], εκτελεί ελέγχους κανόνων επιλύοντας συστήματα περιορισμού με εισόδους που θα κατευθύνουν το πρόγραμμα σε όλες τις δυνατές διαδρομές εκτέλεσης. Εάν ενεργοποιηθεί μία κατάσταση σύγκρουσης, τότε μπορεί να ανακτηθούν οι παράμετροι εισόδου που οδήγησαν σε αυτή, δηλαδή όλοι οι κανόνες που ενεργοποιούνται.

Παρόλα αυτά, θα πρέπει να σημειωθεί πως ο εντοπισμός των συγκρούσεων στις παραπάνω εργασίες είναι κεντρικοποιημένος. Δηλαδή, υποθέτουν πως ένα συγκεντρωτικό σύστημα έχει πλήρη γνώση όλων των πολιτικών του IoT. Ένας τέτοιος μηχανισμός δεν είναι επεκτάσιμος καθώς η πολυπλοκότητα του συλλογισμού πάνω στις πιθανές καταστάσεις των συσκευών αυξάνεται εκθετικά  $O(m^n)$  με την εισαγωγή νέων συσκευών, υποθέτοντας ότι υπάρχει ένα οικοσύστημα που απαρτίζεται από  $n$  αντικείμενα και πως  $m$  καταστάσεις αντιστοιχούν σε κάθε έξυπνο αντικείμενο.

## 2.3 Πολυπρακτορικά Συστήματα

Μία από τις κύριες τεχνολογίες πάνω στην οποία βασίζεται η παρούσα διατριβή, είναι τα Πολυπρακτορικά Συστήματα (Multi-Agent Systems; MAS). Τα Πολυπρακτορικά Συστήματα αποτελούνται από πράκτορες λογισμικού, οι οποίοι αποτελούν αυτόνομες οντότητες και είναι ικανές να:

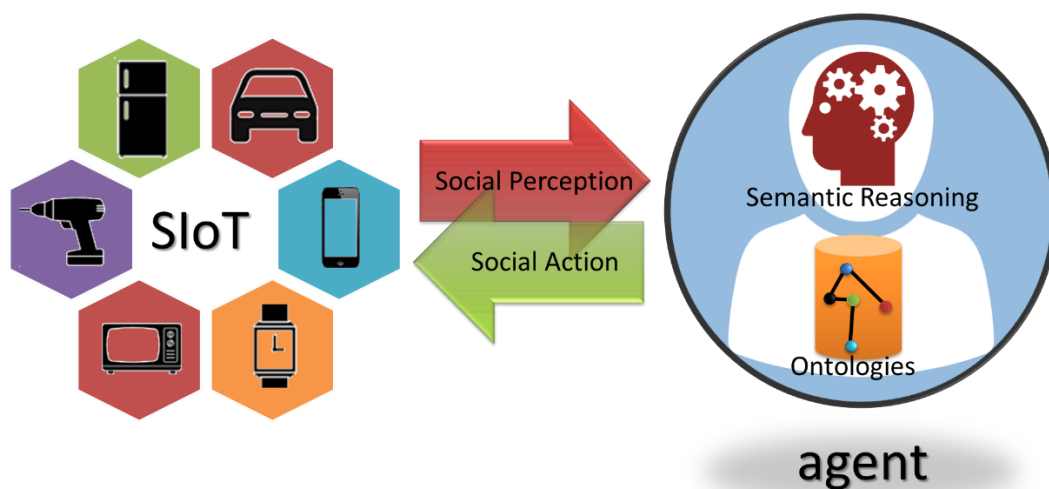
- αντιλαμβάνονται το περιβάλλον (π.χ., να δέχονται δεδομένα και πληροφορίες), στο οποίο ανήκουν και να αντιδρούν έγκαιρα σε αλλαγές που συμβαίνουν σε αυτό,
- επιδεικνύουν συμπεριφορές επικεντρωμένες σε σαφώς ορισμένους στόχους και να λαμβάνουν πρωτοβουλίες κατά περίπτωση,
- αλληλεπιδρούν, όταν είναι απαραίτητο, με άλλους πράκτορες με σκοπό την επίλυση του προβλήματος που τους έχει ανατεθεί ή με σκοπό να βοηθήσουν άλλους πράκτορες με τις δραστηριότητές τους [76].

Σύμφωνα με τα παραπάνω, οι πράκτορες αποτελούν, λοιπόν, την κατάλληλη οντότητα αντιπροσώπευσης των έξυπνων αντικειμένων (κυρίως αυτών που δεν διαθέτουν μεγάλη υπολογιστική ισχύ), αλλά και των ανθρώπων στον κυβερνο-φυσικό χώρο. Ένα επιπλέον πλεονέκτημα που διαθέτουν, είναι η ικανότητά τους να προσαρμόζονται σε δυναμικά περιβάλλοντα [77] και σε ανοιχτά συστήματα, η δομή των οποίων δύναται να μεταβληθεί δυναμικά [78]. Τέτοιο σύστημα/περιβάλλον μπορεί να χαρακτηριστεί το Διαδίκτυο των Πραγμάτων, όπου τα συστατικά του

στοιχεία δεν είναι γνωστά εκ των προτέρων, μπορεί να μεταβληθούν με την πάροδο του χρόνου και συχνά χαρακτηρίζονται από υψηλούς βαθμούς ετερογένειας με την έννοια ότι αναπτύσσονται από διαφορετικούς ανθρώπους, χρησιμοποιώντας διαφορετικές τεχνολογίες, μεθοδολογίες και εργαλεία λογισμικού.

Επιπλέον, η δυναμική του IoT περιβάλλοντος αυξάνει όταν αυτό ακολουθεί το πρότυπο του SIoT, καθώς απαιτείται η διαρκής κοινωνικοποίηση των οντοτήτων που βρίσκονται σε αυτό, δηλαδή ο σχηματισμός σχέσεων φιλίας μεταξύ τους. Ακόμη, η φύση ενός SIoT οικοσυστήματος αυξάνει την ανάγκη για αυτόνομη συμπεριφορά των οντοτήτων του, καθώς πρέπει να διαχειρίζονται τις κοινωνικές τους σχέσεις και να παίρνουν αποφάσεις βασιζόμενες σε αυτές, κάτι που ενισχύει τη χρήση έξυπνων πρακτόρων λογισμικού. Θα πρέπει να σημειωθεί ότι ο σχεδιασμός πολυπρακτορικών συστημάτων προδιαγράφεται από το πρότυπο FIPA (Foundation for Intelligent Physical Agents) [78], και η επικοινωνία μεταξύ τους επιτυγχάνεται μέσω μηνυμάτων, όπως αυτά προδιαγράφονται από το εν λόγω πρότυπο (FIPA ACL message).

Η Εικόνα 11 αποτελεί μία αφαιρετική παρουσίαση της αλληλεπίδρασης ενός πράκτορα λογισμικού που εκπροσωπεί μία οντότητα στο SIoT περιβάλλον.



Εικόνα 11: Αλληλεπίδραση ενός πράκτορα λογισμικού με το SIoT.

## 2.4 Επίγνωση Πλαισίου

Η επίγνωση της κατάστασης και του πλαισίου γίνεται ολοένα και πιο σημαντική στην πορεία προς τη δημιουργία έξυπνων μηχανών και συσκευών, οι οποίες θα προσφέρουν ένα ολοκληρωμένο σύνολο εργαλείων για τη βελτίωση της ποιότητας ζωής μας. Η αυξημένη υπολογιστική ικανότητα των προσωπικών έξυπνων συσκευών και οι συνεχώς αυξανόμενες ικανότητές τους για ανίχνευση, επιτρέπουν την αποθήκευση, επεξεργασία και μετάδοση μεγάλου αριθμού συλλεχθέντων

δεδομένων μέσω κινητών συσκευών και δικτύων. Ως αποτέλεσμα, η γρήγορη και αποτελεσματική επεξεργασία και ανάλυση αυτών των δεδομένων κινητής τηλεφωνίας αποτελεί μία μεγάλη πρόκληση.

### 2.4.1 Ορισμός Επίγνωσης Πλαισίου

Οι συνεχείς εξελίξεις στην επιστήμη των υπολογιστών κατά την τελευταία δεκαετία οδήγησαν στην ανάπτυξη μίας πληθώρας μικρών αισθητήρων που επιτρέπουν στις κινητές συσκευές να αναφέρουν την κατάσταση τους και σε πολλές περιπτώσεις να γνωρίζουν το πλαίσιο στο οποίο βρίσκονται. Ο ορισμός της επίγνωσης πλαισίου πρωτοεμφανίστηκε στην [79] ως *“η ικανότητα των εφαρμογών του κινούμενου χρήστη να ανακαλύψουν και να αντιδράσουν στις αλλαγές του περιβάλλοντος στο οποίο βρίσκονται”*. Αργότερα, οι Abowd και Mynatt [80], καθόρισαν τα πέντε *Ws, Who* (Ποιος), *What* (Τι), *Where* (Πού), *When* (Πότε), και *Why* (Γιατί), ως τις ελάχιστες πληροφορίες που είναι απαραίτητες για την κατανόηση του πλαισίου. Παρόλα αυτά, ο επικρατέστερος ορισμός δόθηκε από τον Dey [81], ο οποίος θέλοντας να δώσει μία ευρύτερη ερμηνεία καθόρισε την επίγνωση πλαισίου ως *“κάθε πληροφορία που μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει την κατάσταση μίας οντότητας, όπου μία οντότητα μπορεί να είναι ένα άτομο, μία τοποθεσία, ένα φυσικό ή ένα υπολογιστικό αντικείμενο”*.

Τα συστήματα που έχουν επίγνωση πλαισίου εκμεταλλεύονται μικρούς ενσωματωμένους αισθητήρες σε κινητές και φορητές συσκευές, προκειμένου να αντιληφθούν ατομικά ή ομαδικά μοτίβα. Ως αποτέλεσμα, οι προγραμματιστές είναι σε θέση να αναπτύξουν προηγμένες εφαρμογές που έχουν επίγνωση πλαισίου για τον εντοπισμό των φυσικών και των κοινωνικών δραστηριοτήτων των χρηστών. Οι εφαρμογές που έχουν επίγνωση πλαισίου στοχεύουν στην ενθάρρυνση των χρηστών να μοιράζονται τα δεδομένα των αισθητήρων τους, σε μία προσπάθεια δημιουργίας ενός γνωσιακού κοινωνικού δικτύου.

Το σημαντικό χαρακτηριστικό, το οποίο προσφέρεται μέσω της επεξεργασίας δεδομένων που συλλέγονται από φορητές ή προσωπικές συσκευές με δυνατότητες ανίχνευσης, είναι η συγχώνευση των δεδομένων (π.χ., δεδομένων επιταχυνσιόμετρου) έχοντας ως στόχο την εξαγωγή πληροφορίας σχετιζόμενης με το πλαίσιο ή την κατάσταση στην οποία βρίσκεται ο χρήστης (π.χ., ανεβαίνει σκάλες). Η ανάλυση των ανθρωποκεντρικών μοτίβων όχι μόνο μπορεί να βελτιώσει την ποιότητα της ζωής των ατόμων αλλά μέσω της χρήσης της σε ένα συλλογικό πλαίσιο, όπως ένα κοινωνικό δίκτυο, μπορεί να εκμεταλλευθεί αυτή τη γνώση για να συναγάγει ομαδικά μοτίβα, προκειμένου να λάβει αποφάσεις για το κοινό καλό. Για παράδειγμα, εάν παρατηρηθεί μία ξαφνική αλλαγή στην κίνηση μίας ομάδας ανθρώπων (π.χ., μία ομάδα ατόμων που βρίσκονται στην ίδια τοποθεσία αρχίζει να τρέχει) μπορεί να είναι η ένδειξη μίας επικίνδυνης κατάστασης (π.χ., τρομοκρατική επίθεση). Παρόμοιες



πληροφορίες μπορεί να εξαχθούν από τις αναρτήσεις σε ένα κοινωνικό δίκτυο, κάνοντας χρήση τεχνικών επεξεργασίας φυσικής γλώσσας.

Οι εφαρμογές επίγνωσης πλαισίου μπορεί να ομαδοποιηθούν σε δύο διαφορετικές κατηγορίες: α) προσωπικές/ανθρωποκεντρικές και β) αστικές, οι οποίες χωρίζονται σε συμμετοχικές/ομαδικές ή ευκαιριακές [82]. Οι εφαρμογές του πρώτου τύπου (ανθρωποκεντρικές εφαρμογές) αφορούν το χρήστη. Για παράδειγμα, μία εφαρμογή αναγνώρισης δραστηριότητας ανακαλύπτει μοτίβα κίνησης ενός ατόμου αναλύοντας ιστορικά δεδομένα αισθητήρων (π.χ., δεδομένα επιταχυνσιόμετρου), για να ταξινομήσει την τρέχουσα δραστηριότητά του. Από την άλλη πλευρά, η συλλογική παρακολούθηση δίνει έμφαση σε πολλαπλές εφαρμογές εγκατεστημένες σε φορητές/φορετές συσκευές που συλλέγουν, μοιράζονται και αναλύουν τοπικά τις γνώσεις βασισμένες στις ανθρώπινες δραστηριότητες και στο περιβάλλον. Επομένως, η συλλογική επίγνωση πλαισίου, η οποία επιτυγχάνεται μέσω των εφαρμογών της δεύτερης κατηγορίας, απαιτεί τη συμμετοχή πολλών χρηστών για την παροχή δεδομένων αισθητήρων, σε μία προσπάθεια να ανακαλυφθούν φαινόμενα/μοτίβα μεγάλης κλίμακας. Για παράδειγμα, τα δεδομένα σχετικά με την τοποθεσία και την ταχύτητα των οδηγών/χρηστών ενδέχεται να βοηθήσουν στη σύνταξη μίας έξυπνης εφαρμογής για την κυκλοφοριακή συμφόρηση. Τέλος, σε αντίθεση με τις δύο προηγούμενες μορφές ανίχνευσης, η ευκαιριακή ανίχνευση συλλέγει δεδομένα αισθητήρων χωρίς την ενεργή αλληλεπίδραση του χρήστη. Πιο συγκεκριμένα, πρόκειται για εφαρμογές που τρέχουν στο παρασκήνιο (background), και ενεργοποιούνται εφόσον υπάρξει κάποια σημαντική μεταβολή στα δεδομένα των αισθητήρων. Για παράδειγμα, αν ακουσθεί ένας πολύ δυνατός θόρυβος (π.χ., πυροβολισμός) η εφαρμογή μπορεί να ανοίξει την κάμερα (χωρίς τη συμμετοχή του χρήστη) και να τραβήξει κάποιες φωτογραφίες.

Οι πληροφορίες επίγνωσης πλαισίου έχουν κατηγοριοποιηθεί βάσει των οντοτήτων/πόρων που παρέχουν τα δεδομένα (τοποθεσίες, άτομα και αντικείμενα), σε τέσσερις κατηγορίες [83]:

1. Ταυτότητα (π.χ., κάνοντας χρήση ενός αισθητήρα GPS),
2. Τοποθεσία (π.χ., κάνοντας χρήση μίας ετικέτας RFID),
3. Κατάσταση (π.χ., μία έξυπνη λάμπα είναι σε κατάσταση On ή Off), και
4. Χρόνο (π.χ., κάνοντας χρήση ενός χρονόμετρου).

Οι παραπάνω κατηγορίες χαρακτηρίστηκαν από την [81] ως πρωτογενείς πληροφορίες πλαισίου, γιατί μπορεί να ανακτηθούν κατευθείαν από την πηγή (αισθητήρα), ενώ πληροφορίες που προκύπτουν μέσω συγχώνευσης δεδομένων (data fusion) αισθητήρων. Πιο συγκεκριμένα:

- *Πρωτογενείς πληροφορίες πλαισίου*: Οποιαδήποτε πληροφορία ανακτάται χωρίς να χρησιμοποιείται το υπάρχον πλαίσιο και χωρίς να εκτελείται οποιαδήποτε λειτουργία συγχώνευσης δεδομένων (data fusion) αισθητήρων (π.χ., αναγνώσεις αισθητήρων GPS ως πληροφορίες θέσης).
- *Δευτερογενείς πληροφορίες πλαισίου*: Οποιοσδήποτε πληροφορίες μπορούν να υπολογισθούν χρησιμοποιώντας τις πρωτογενείς πληροφορίες πλαισίου. Οι δευτερογενείς πληροφορίες πλαισίου μπορεί να υπολογισθούν χρησιμοποιώντας λειτουργίες συγχώνευσης δεδομένων αισθητήρων ή λειτουργίες ανάκτησης δεδομένων όπως η κατανάλωση μίας υπηρεσίας ιστού (π.χ., η ανάκτηση της λίστας των φίλων του χρήστη, μέσω της Facebook API).

Η ανάκτηση των δευτερογενών πληροφοριών πλαισίου κατάστασης μέσω φορητών/κινητών συσκευών σε πολλές περιπτώσεις είναι εξαιρετικά δύσκολη διότι απαιτούνται τεχνικές μηχανικής μάθησης. Για παράδειγμα, η αναγνώριση της κατάστασης στην οποία βρίσκεται ο χρήστης (π.χ., χρησιμοποιώντας δεδομένα τριαξονικού επιταχυνσιόμετρου). Στην επόμενη ενότητα, επισημαίνονται οι πιο συνηθισμένες εφαρμογές επίγνωσης πλαισίου που σχετίζονται με την αναγνώριση κατάστασης του χρήστη (σωματικής και ψυχολογικής).

#### **2.4.2 Εφαρμογές επίγνωσης πλαισίου με χρήση φορητών συσκευών**

Η πληροφόρηση σχετικά με τη σωματική δραστηριότητα ενός χρήστη μπορεί να χρησιμοποιηθεί για την ενίσχυση της ποιότητας της ζωής του, καθώς και από εφαρμογές έγκαιρης προειδοποίησης της κατάστασης της υγείας του ατόμου. Επιπλέον, οι εφαρμογές που αφορούν την πολιτική προστασία μπορούν να βελτιώσουν την κατάσταση ετοιμότητας των φορέων επιβολής του νόμου, οι οποίοι θα μπορούσαν να μειώσουν δραματικά το χρόνο απόκρισής τους σε κρίσιμες καταστάσεις. Υπάρχουν αρκετές εφαρμογές που βασίζονται στην επίγνωση πλαισίου που επικεντρώνεται κυρίως σε μεμονωμένα γεγονότα, αλλά μπορεί επίσης να εφαρμοστούν συλλογικά για να ανακαλύψουν πληροφορίες σχετικές με γεγονότα που αφορούν μία κοινότητα.

Η *Αναγνώριση Ανθρώπινης Δραστηριότητας (ΑΑΔ)* είναι μία εφαρμογή κατηγοριοποίησης, η οποία επεξεργάζεται δεδομένα που παράγονται από κινητούς/φορητούς αισθητήρες, τους οποίους μεταφέρει ένας χρήστης μαζί του προκειμένου να αναγνωρίσει η εφαρμογή τη φυσική δραστηριότητα που ασκεί [84]. Τα τριαξονικά επιταχυνσιόμετρα είναι οι συνηθέστερα χρησιμοποιούμενοι ενσωματωμένοι αισθητήρες για την αναγνώριση των ανθρώπινων δραστηριοτήτων (π.χ., περπάτημα, τρέξιμο κ.λπ.).

Η *αναγνώριση χειρονομίας* έχει ως στόχο να αναγνωρίσει τις κινήσεις του χεριού ή του βραχίονα ενός υποκειμένου [85]. Οι χειρονομίες χειρός/βραχίονα ενδέχεται να υποδηλώνουν δραστηριότητες, τις οποίες μία εφαρμογή ΑΑΔ δυσκολεύεται να ανακαλύψει. Για παράδειγμα, οι εφαρμογές αναγνώρισης χειρονομίας μπορούν να εντοπίσουν εάν το υποκείμενο χαιρετά ή καπνίζει ένα τσιγάρο.

Οι εφαρμογές *ανίχνευσης πτώσης* αναγνωρίζουν αν έχει πέσει το υποκείμενο που φέρει μαζί του ένα κινητό τηλέφωνο ή μία φορητή συσκευή [86]. Πρόκειται για μία ευρέως χρησιμοποιούμενη εφαρμογή με επίγνωση πλαισίου στον τομέα της Υποβοηθούμενης Αυτόνομης Διαβίωσης (Ambient Assisted Living AAL). Επειδή μία πτώση για τους ηλικιωμένους μπορεί να έχει ως συνέπεια τη νοσηλεία τους και σε ορισμένες περιπτώσεις ίσως ακόμη και να αποβεί μοιραία, η ταχεία απόκριση δύναται να αντιμετωπίσει αποτελεσματικότερα αυτές τις αρνητικές συνέπειες και η χρήση εφαρμογών ανίχνευσης πτώσης συμβάλλει προς αυτή την κατεύθυνση.

Η *ανίχνευση σύγκρουσης* έχει ως στόχο την παρακολούθηση των ατυχημάτων μεταξύ οχημάτων. Η άμεση αντίληψη των τροχαίων ατυχημάτων μειώνει τον αριθμό των θανατηφόρων ελαττώνοντας το χρόνο απόκρισης των ανταποκριτών έκτακτης ανάγκης. Η ανίχνευση σύγκρουσης από κινητές συσκευές προσφέρει πολλά πλεονεκτήματα σε σύγκριση με τα συμβατικά συστήματα ανίχνευσης ατυχημάτων που είναι ενσωματωμένα στο όχημα, καθώς είναι ανεξάρτητες από το όχημα [87].

Η *αντίληψη των περιβαλλοντικών συνθηκών* παρέχει πληροφορίες πλαισίου που σχετίζονται με το περιβάλλον του χρήστη. Τα περιβαλλοντικά χαρακτηριστικά όπως η θερμοκρασία, η υγρασία και η ένταση του φωτισμού του περιβάλλοντος δεν παρέχουν πολλές πληροφορίες όταν χρησιμοποιούνται ξεχωριστά. Από την άλλη πλευρά, ένας συνδυασμός αυτών, όπως για παράδειγμα οι τιμές της πίεσης του αέρα σε συνδυασμό με τις μετρήσεις της έντασης του φωτός, μπορεί να αποκαλύψουν εάν ένα άτομο είναι σε εξωτερικό ή σε εσωτερικό χώρο [88].

Οι εφαρμογές *αναγνώρισης στρες* χρησιμοποιούνται για να ανιχνεύσουν εάν το άτομο είναι υπό πίεση. Ωστόσο, το άγχος είναι μία πολύ εξατομικευμένη βιολογική αντίδραση και, ως εκ τούτου, η εφαρμογή μοντέλων αναγνώρισης του άγχους σε μεγαλύτερες ομάδες ανθρώπων παραμένει ένα ανοιχτό ζήτημα [89].

Η *αναγνώριση συναισθημάτων* υπόσχεται να ανακαλύψει την τρέχουσα συναισθηματική κατάσταση ενός ατόμου. Σε πρόσφατη εργασία [90], επιτυγχάνεται 50% ακρίβεια αναγνώρισης συναισθημάτων μεταξύ τεσσάρων τάξεων (λύπη, χαρά, φόβος, θυμός) μέσω της ανάλυσης των δεδομένων μίας φορητής συσκευής ηλεκτροκαρδιογραφήματος (Electrocardiography; ECG). Ωστόσο, τα αληθινά θετικά αποτελέσματα της κατηγορίας του φόβου επιτυγχάνονται με ακρίβεια μόλις 25%, καθιστώντας τα ίσα με τυχαίες προβλέψεις.

Ο Πίνακας 1 συνοψίζει τις εφαρμογές που έχουν επίγνωση του πλαισίου και παρουσιάζει τον κινητό/φορητό αισθητήρα στον οποίο βασίζονται. Αν και τα παραπάνω παραδείγματα αναφέρονται στην αναγνώριση καταστάσεων ατόμων, ο συνδυασμός συλλεγόμενων πληροφοριών μπορεί να αποκαλύψει περισσότερα

**Πίνακας 1: Αισθητήρες κινητών/φορητών συσκευών που χρησιμοποιούνται σε εφαρμογές επίγνωσης πλαισίου**

Εφαρμογή	Τύπος αισθητήρα
Αναγνώριση Ανθρώπινης Δραστηριότητας	<ul style="list-style-type: none"> <li>• Επιταχυνσιόμετρο</li> <li>• Γυροσκόπιο</li> <li>• Μαγνητόμετρο</li> <li>• Καρδιακός Παλμός</li> </ul>
Αναγνώριση Χειρονομίας	<ul style="list-style-type: none"> <li>• Επιταχυνσιόμετρο</li> <li>• Γυροσκόπιο</li> </ul>
Ανίχνευση πτώσης	<ul style="list-style-type: none"> <li>• Επιταχυνσιόμετρο</li> <li>• Γυροσκόπιο</li> <li>• Αισθητήρας Εγγύτητας</li> </ul>
Ανίχνευση Σύγκρουσης	<ul style="list-style-type: none"> <li>• Επιταχυνσιόμετρο</li> <li>• GPS</li> </ul>
Αντίληψη Περιβαλλοντικών Συνθηκών	<ul style="list-style-type: none"> <li>• Φωτεινότητα</li> <li>• Υγρασία</li> <li>• Θερμοκρασία</li> <li>• Πίεση</li> <li>• GPS</li> </ul>
Αναγνώριση Στρες	<ul style="list-style-type: none"> <li>• Μικρόφωνο</li> <li>• Ηλεκτροκαρδιογράφημα</li> <li>• Καρδιακός παλμός</li> <li>• Δερματική αγωγιμότητα</li> </ul>
Αναγνώριση Συναισθημάτων	<ul style="list-style-type: none"> <li>• Μικρόφωνο</li> <li>• Κάμερα</li> <li>• Ηλεκτροκαρδιογράφημα</li> <li>• Καρδιακός παλμός</li> <li>• Δερματική αγωγιμότητα</li> </ul>

σχετικά με κοινωνικά γεγονότα ή καταστάσεις, με βάση τη συλλογική γνώση που εξάγεται. Για παράδειγμα, αν και οι υψηλές μετρήσεις στάθμης τάσης σε έναν μεμονωμένο χρήστη μπορεί να υποδεικνύουν ότι ο συγκεκριμένος χρήστης είναι ενδεχομένως σε επικίνδυνη κατάσταση, παρόμοιες μετρήσεις αρκετών ατόμων στην ίδια περιοχή θα μπορούσαν να υποδηλώνουν ένα επικίνδυνο γεγονός που απειλεί τη δημόσια ασφάλεια.

### 2.4.3 Στάδια Εξαγωγής και Επεξεργασίας Πληροφοριών Πλαισίου

Η συλλογιστική του πλαισίου (context reasoning) μπορεί να οριστεί ως μία παραγωγική μέθοδος εξαγωγής νέων γνώσεων που προσφέρει καλύτερη κατανόηση των δεδομένων, με βάση το διαθέσιμο πλαίσιο. Μπορεί επίσης να εξηγηθεί ως μία διαδικασία παροχής παραγωγικών συμπερασμάτων πλαισίου υψηλού επιπέδου από ένα σύνολο πλαισίων [91]. Επιπρόσθετα, η απαίτηση της συλλογιστικής προέκυψε λόγω δύο χαρακτηριστικών του ακατέργαστου περιβάλλοντος: ατέλεια (δηλαδή διφορούμενων, ασαφών ή λανθασμένων μετρήσεων) και αβεβαιότητα.

Για την καλύτερη κατανόηση της διαφοράς μεταξύ των ακατέργαστων δεδομένων και των πληροφοριών πλαισίου ο Sanchez και λοιποί [92] εξήγησαν τη διάκρισή τους ως εξής:

- Ακατέργαστα δεδομένα (αισθητήρα): Είναι μη επεξεργασμένα και ανακτώνται απευθείας από την πηγή δεδομένων, όπως είναι οι αισθητήρες.
- Πληροφορίες πλαισίου: Δημιουργούνται με την επεξεργασία ακατέργαστων δεδομένων αισθητήρων. Επιπλέον, ελέγχονται ως προς τη συνοχή τους και προστίθενται σε αυτές μεταδεδομένα.

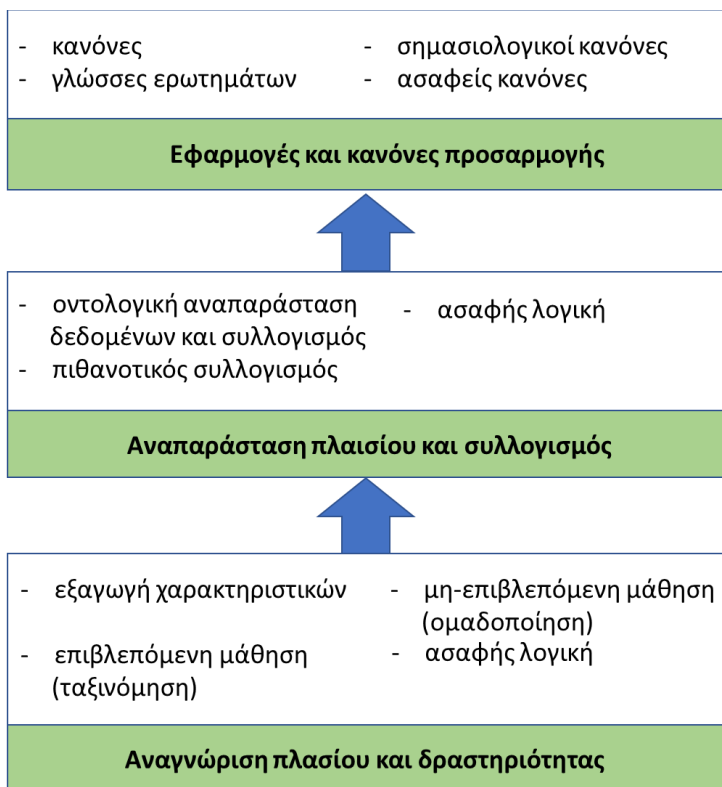
Η απόδοση του συλλογισμού μπορεί να μετρηθεί χρησιμοποιώντας ως μετρικές την αποτελεσματικότητα, την ορθότητα, την πληρότητα και τη διαλειτουργικότητα [93]. Η συλλογιστική του πλαισίου περιλαμβάνει διάφορα βήματα. Σε γενικές γραμμές μπορεί να διαχωριστεί σε τρεις φάσεις [94]:

- *Προεπεξεργασία Πλαισίου*: ασχολείται με τον καθαρισμό (π.χ., κάποια δεδομένα μπορεί να λείπουν ή να μην είναι ακριβή) των σημάτων των αισθητήρων.
- *Συγχώνευση Δεδομένων Αισθητήρων*: πρόκειται για μία μέθοδο συνδυασμού δεδομένων, προερχόμενα από πολλούς αισθητήρες, έχοντας ως στόχο την παραγωγή ακριβέστερων, πληρέστερων και πιο αξιόπιστων πληροφοριών που δεν θα μπορούσε να επιτευχθεί μέσω ενός μόνο αισθητήρα.
- *Συμπερασμός Πλαισίου*: πρόκειται για τη δημιουργία πληροφοριών πλαισίου υψηλού επιπέδου χρησιμοποιώντας πληροφορίες πλαισίου χαμηλότερου επιπέδου.

Οι πληροφορίες πλαισίου και οι συναφείς με αυτές γνώσεις εκπροσωπούνται και επεξεργάζονται σε σύγχρονα συστήματα που έχουν επίγνωση του πλαισίου. Η Εικόνα 12 απεικονίζει τα τρία επίπεδα εξαγωγής και επεξεργασίας πληροφοριών πλαισίου

παρουσιάζοντας κάποιες χρησιμοποιούμενες μεθόδους και τεχνικές σε κάθε στρώμα [93]. Τα βέλη απεικονίζουν τη ροή πληροφοριών πλαισίου.

Στο κάτω στρώμα εφαρμόζονται τεχνικές επεξεργασίας σήματος και μηχανικής μάθησης για την αναγνώριση του πλαισίου και των δραστηριοτήτων από σήματα αισθητήρων και από άλλες πηγές ακατέργαστων δεδομένων. Το μεσαίο στρώμα είναι ο πυρήνας της μοντελοποίησης των πληροφοριών πλαισίου, όπου έχουν εφαρμοστεί κατά το παρελθόν πολλαπλές τεχνικές αναπαράστασης και συλλογισμού. Οι εξαιρετικά δυναμικές έξοδοι του κάτω στρώματος θέτουν πολλές απαιτήσεις στο μεσαίο στρώμα, όσον αφορά τη διαχείριση της γνώσης. Τέλος, το επάνω στρώμα ασχολείται με τη διάδοση, των εξαρτωμένων από το πλαίσιο, ενεργειών ή της προσαρμογής των αντικειμένων με βάση τις πληροφορίες πλαισίου (δηλ., μέσω της χρήσης κανόνων). Οι εφαρμογές που αναπτύσσονται πρέπει να χρησιμοποιούν μία γλώσσα ερωτήματος πλαισίου για την πρόσβαση σε πληροφορίες πλαισίου από τις βάσεις δεδομένων ή τους παρόχους υπηρεσιών. Οι γλώσσες ερωτημάτων μπορούν να χρησιμοποιηθούν για να περιγράψουν ερωτήματα καθώς και συνδρομές. Ο σχεδιασμός ή η επιλογή της γλώσσας ερωτήματος εξαρτάται από τις τεχνικές αναπαράστασης που χρησιμοποιούνται στο μεσαίο στρώμα. Ένας σημαντικός ρόλος του μεσαίου στρώματος και της γλώσσας ερωτήματος είναι να εξαλειφθεί η άμεση σύνδεση των πηγών πλαισίου που παρέχει τα ακατέργαστα δεδομένα με τα συστατικά στοιχεία λογισμικού που «καταναλώνουν» το πλαίσιο.



Εικόνα 12: Στάδια εξαγωγής και επεξεργασίας πληροφοριών πλαισίου

Ο Πίνακας 2 παρουσιάζει τα πλεονεκτήματα και τα μειονεκτήματα των τεχνολογιών που χρησιμοποιούνται για την εξαγωγή και την επεξεργασία πληροφοριών πλαισίου, καθώς και ένα παράδειγμα εφαρμογής της κάθε τεχνικής [21].

**Πίνακας 2: Τεχνικές εξαγωγής και επεξεργασίας πληροφοριών πλαισίου**

Τεχνικές	Θετικά	Αρνητικά	Εφαρμογές
<p><b>Επιβλεπόμενη Μάθηση</b></p> <p>(Αλγόριθμος k πλησιέστερου γείτονα, Μηχανές Διανυσμάτων Υποστήριξης, Δέντρα Αποφάσεων, Τυχαία Δάση, Τεχνητά Νευρωνικά Δίκτυα κτλ.)</p>	<ul style="list-style-type: none"> <li>• Αρκετά ακριβής</li> <li>• Μεγάλος αριθμός εναλλακτικών μοντέλων</li> <li>• Στηρίζεται στη χρήση μαθηματικών και στατιστικά συναρτήσεων</li> <li>• Κατάλληλη για επεξεργασία αριθμητικών δεδομένων</li> </ul>	<ul style="list-style-type: none"> <li>• Απαιτείται σημαντική ποσότητα δεδομένων</li> <li>• Κάθε στοιχείο δεδομένων πρέπει να μετατρέπεται σε αριθμητικές τιμές</li> <li>• Η επιλογή των χαρακτηριστικών μπορεί να είναι απαιτητική</li> <li>• Μπορεί να είναι πιο έντονη (επεξεργασία, αποθήκευση, χρόνος)</li> <li>• Λιγότερο σημασιολογικά μοντέλα</li> <li>• Απαιτούνται χαρακτηρισμένα δεδομένα εκπαίδευσης</li> <li>• Τα μοντέλα μπορεί να είναι πολύπλοκα</li> <li>• Δύσκολο να καταγράψουν την υπάρχουσα γνώση</li> </ul>	<p>Για περιπτώσεις όπου το σύνολο χαρακτηριστικών είναι εύκολα αναγνωρίσιμο, τα πιθανά αποτελέσματα είναι γνωστά, και μεγάλα σύνολα αριθμητικών δεδομένων (για εκπαίδευση επίσης) είναι διαθέσιμα. (Π.χ., αναγνώριση ανθρώπινης δραστηριότητας, αναγνώριση ελλিপών τιμών)</p>
<p><b>Μη-επιβλεπόμενη Μάθηση</b></p> <p>(Ομαδοποίηση, αλγόριθμος k-means κτλ.)</p>	<ul style="list-style-type: none"> <li>• Δεν απαιτούνται χαρακτηρισμένα δεδομένα εκπαίδευσης</li> <li>• Στηρίζεται στη χρήση μαθηματικών και στατιστικά συναρτήσεων</li> <li>• Κατάλληλη για επεξεργασία αριθμητικών δεδομένων</li> </ul>	<ul style="list-style-type: none"> <li>• Κάθε στοιχείο δεδομένων πρέπει να αναπαρίσταται σε αριθμητικές τιμές</li> <li>• Μπορεί να είναι πιο έντονη (επεξεργασία, αποθήκευση, χρόνος)</li> <li>• Λιγότερο σημασιολογικά μοντέλα</li> <li>• Τα μοντέλα μπορεί να είναι πολύπλοκα</li> <li>• Τα επιθυμητά αποτελέσματα δεν είναι γνωστά</li> </ul>	<p>Για καταστάσεις όπου δεν είναι γνωστά τα πιθανά αποτελέσματα (π.χ., ανίχνευση ασυνήθιστων συμπεριφορών, ανάλυση αγροτικών πεδίων για τον εντοπισμό κατάλληλης θέσης για την εγκατάσταση συγκεκριμένου είδους καλλιέργειας)</p>
<p><b>Γλώσσες Κανόνων</b></p>	<ul style="list-style-type: none"> <li>• Απλός ορισμός</li> <li>• Επεκτασιμότητα</li> <li>• Λιγότεροι πόροι (π.χ., επεξεργασία, αποθήκευση)</li> </ul>	<ul style="list-style-type: none"> <li>• Πρέπει να ορίζονται με μη αυτόματο τρόπο</li> <li>• Μπορεί να είναι επιρρεπής σε σφάλματα λόγω ανθρώπινου παράγοντα</li> <li>• Δεν υπάρχει επικύρωση ή έλεγχος ποιότητας</li> </ul>	<p>Για καταστάσεις όπου τα στοιχεία των ακατέργαστων δεδομένων πρέπει να μετατραπούν σε πληροφορίες περιβάλλοντος υψηλού επιπέδου. Κατάλληλο για χρήση για τον ορισμό συμβάντων.</p>
<p><b>Ασαφής Λογική</b></p>	<ul style="list-style-type: none"> <li>• Επιτρέπει μία πιο φυσική εκπροσώπηση των τιμών</li> <li>• Απλός ορισμός</li> <li>• Επεκτασιμότητα</li> </ul>	<ul style="list-style-type: none"> <li>• Πρέπει να ορίζεται με μη αυτόματο τρόπο</li> </ul>	<p>Για καταστάσεις όπου χρειάζονται χαμηλού επιπέδου πληροφορίες να μετατραπούν σε υψηλού επιπέδου πιο κατανοητές πληροφορίες</p>

	<ul style="list-style-type: none"> <li>• Λιγότεροι πόροι (π.χ., επεξεργασία, αποθήκευση)</li> <li>• Μπορεί να χειριστεί την αβεβαιότητα</li> </ul>	<ul style="list-style-type: none"> <li>• Μπορεί να είναι επιρρεπής σε σφάλματα λόγω ανθρώπινου παράγοντα</li> <li>• Δεν υπάρχει επικύρωση ή έλεγχος ποιότητας</li> <li>• Μπορεί να μειώσει την ποιότητα (π.χ., ακρίβεια) των αποτελεσμάτων λόγω φυσικών αναπαραστάσεων</li> </ul>	<p>πλαisiού. Αυτό το είδος απλοποίησης θα διευκολύνει την περαιτέρω επεξεργασία τους. Π.χ., στην περίπτωση της αυτόματης άρδευσης όπου το νερό θα απελευθερωθεί όταν το σύστημα ανίχνευσης του εδάφους θεωρήσει το περιβάλλον ως «ξηρό».</p>
<p><b>Οντολογίες (Λογική Πρώτου Βαθμού)</b></p>	<ul style="list-style-type: none"> <li>• Επιτρέπει περίπλοκο συλλογισμό</li> <li>• Επιτρέπει περίπλοκη αναπαράσταση των δεδομένων</li> <li>• Πιο κατανοητά προς τον άνθρωπο αποτελέσματα</li> <li>• Είναι δυνατός ο έλεγχος επικύρωσης και ποιότητας</li> <li>• Μπορεί να χειριστεί τόσο τα αριθμητικά δεδομένα όσο και τα κείμενα</li> </ul>	<ul style="list-style-type: none"> <li>• Τα δεδομένα πρέπει να διαμορφώνονται σε συμβατή μορφή (π.χ., OWL, RDF)</li> <li>• Περιορισμένη αριθμητική συλλογιστική</li> <li>• Μπορεί να είναι πιο έντονη (επεξεργασία, αποθήκευση, χρόνος)</li> </ul>	<p>Για καταστάσεις όπου η γνώση είναι σημαντική. Π.χ., να αποθηκευθούν και να ερμηνευθούν τα δεδομένα στον τομέα των γεωργικών τομέων. Επιτρέπει την αποθήκευση πληροφοριών περιεχομένου σύμφωνα με τη δομή οντολογιών και αυτόματα εφαρμόζει λογική πρώτου βαθμού με βάση την οντολογία.</p>
<p><b>Πιθανοτικός Λογισμός</b> (Κρυφά Μαρκοβιανά Μοντέλα, Bayesian Δίκτυα κτλ.)</p>	<ul style="list-style-type: none"> <li>• Επιτρέπει το συνδυασμό αποδεικτικών στοιχείων</li> <li>• Μπορεί να χειριστεί αθέατες καταστάσεις</li> <li>• Υπάρχουν διαθέσιμα εναλλακτικά μοντέλα</li> <li>• Μπορεί να χειριστεί την αβεβαιότητα</li> <li>• Παρέχει σχετικά κατανοητά αποτελέσματα</li> </ul>	<ul style="list-style-type: none"> <li>• Πρέπει να είναι γνωστές οι πιθανότητες</li> <li>• Μπορεί να επεξεργαστεί μόνο αριθμητικά ορισμένα δεδομένα</li> </ul>	<p>Για καταστάσεις όπου οι πιθανότητες είναι γνωστές και ο συνδυασμός στοιχείων από διαφορετικές πηγές είναι απαραίτητος. Για παράδειγμα, στοιχεία που παράγονται από μία κάμερα, αισθητήρες υπέρυθρης ακτινοβολίας, έναν αισθητήρα ακουστικής και έναν ανιχνευτή κίνησης σε συνδυασμό για την ανίχνευση διείσδυσης άγριων ζώων σε ένα καλλιεργήσιμο χωράφι.</p>

## 2.5 Επεξεργασία Σημάτων Κίνησης

Η ανάλυση των δεδομένων κίνησης έχοντας ως στόχο την αναγνώριση των ανθρώπινων δραστηριοτήτων, και συνεπώς την επίγνωση πλαisiού, είναι ένας ερευνητικός κλάδος της επεξεργασίας σήματος (signal processing). Η επεξεργασία των σημάτων κίνησης επεξεργάζεται τις ροές δεδομένων ως σήματα και εκμεταλλεύεται μαθηματικές, στατιστικές και ευρετικές αναπαραστάσεις προκειμένου να αντλήσει πολύτιμες πληροφορίες από αυτές. Η πιο συνηθισμένη διαδικασία επεξεργασίας σήματος παρουσιάζεται στην Εικόνα 13, ενώ όλα τα βήματα (συλλογή ακατέργαστων δεδομένων, προεπεξεργασία, τμηματοποίηση, εξαγωγή χαρακτηριστικών και κατηγοριοποίηση) περιγράφονται παρακάτω.





Εικόνα 13: Η συνήθης διαδικασία επεξεργασίας σήματος [14].

### 2.5.1 Συλλογή ακατέργαστων δεδομένων

Τα δεδομένα που συλλέγονται από τους αισθητήρες των κινητών είναι σε μορφή χρονοσειράς αλλά είναι ακατέργαστα, συνεπώς δεν παρουσιάζονται σε δομημένη μορφή. Ως αποτέλεσμα, μπορεί να λείπουν κάποιες τιμές δεδομένων (π.χ., εξαιτίας της απώλειας επικοινωνίας μεταξύ πομπού και δέκτη), ενώ, επειδή η μέγιστη συχνότητα δειγματοληψίας με την οποία ένας αισθητήρας αποδίδει τα δεδομένα διαφέρει ανάλογα με την κινητή συσκευή, για να αναλυθούν οι χρονοσειρές απαιτείται το στάδιο της προεπεξεργασίας.

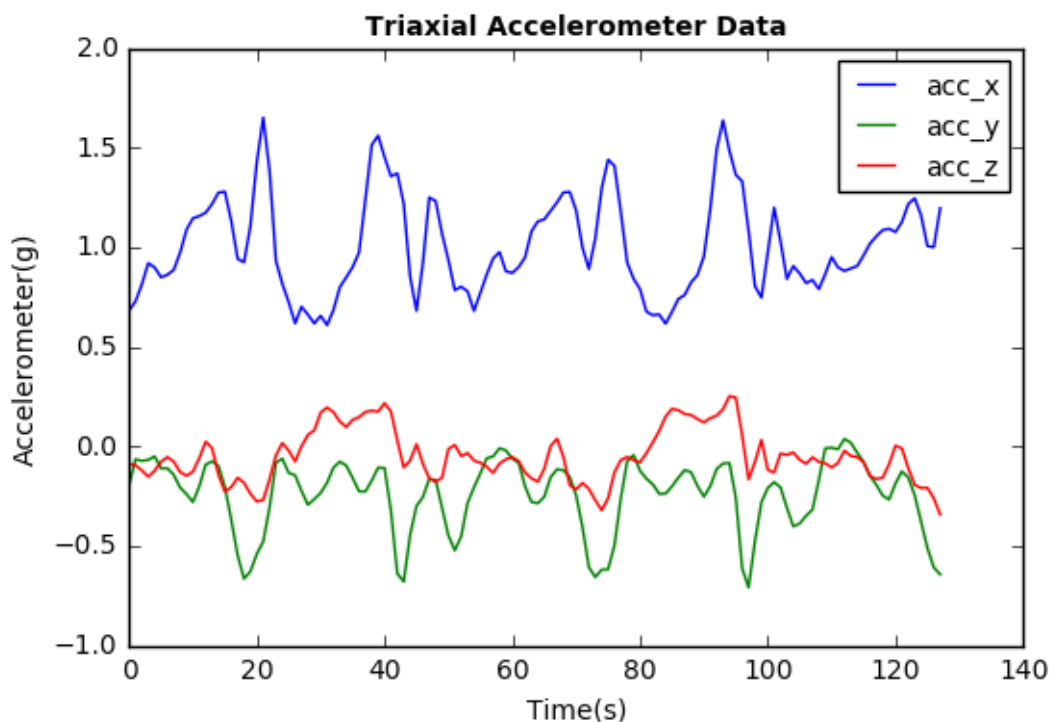
### 2.5.2 Προεπεξεργασία δεδομένων

Σε αυτό το στάδιο, τα ακατέργαστα δεδομένα αισθητήρων από τις φορητές/κινητές συσκευές συγχρονίζονται, τιτλοφορούνται και σημειώνονται χρονικά στο στάδιο της προεπεξεργασίας. Το βήμα προεπεξεργασίας ασχολείται επίσης με τις τιμές δεδομένων που λείπουν. Μία απλή μέθοδος χειρισμού των ελλειπουσών τιμών είναι η γραμμική παρεμβολή, η οποία μπορεί να αντικατασταθεί από πιο σύνθετες μεθόδους [95]. Επιπλέον, προκειμένου να αποφευχθεί η αντιμετώπιση ενδεχόμενων μεταβατικών δραστηριοτήτων στην αναγνώριση ανθρώπινης δραστηριότητας, διαγράφονται 10 δευτερόλεπτα από την αρχή και το τέλος κάθε επισημασμένης δραστηριότητας [96].

### 2.5.3 Τμηματοποίηση

Η τμηματοποίηση δεδομένων αναφέρεται στη διαδικασία διαίρεσης ενός συνόλου δεδομένων σε μικρότερα σύνολα. Στην επεξεργασία σήματος, οι ερευνητές πρέπει να ορίσουν ένα μήκος του χρονικού παραθύρου που πρόκειται να χρησιμοποιήσουν. Για παράδειγμα, εάν ένας αισθητήρας παράγει δεδομένα με συχνότητα δειγματοληψίας 50 Hz και το επιλεγμένο χρονικό παράθυρο ίσο με 2 δευτερόλεπτα, αυτό έχει ως αποτέλεσμα την ύπαρξη 100 χρονικά επισημασμένων

τιμών ανά δείγμα δεδομένων. Σύμφωνα με προηγούμενη έρευνα ([97]), στην ΑΑΔ δεν υπάρχει κάποιο βέλτιστο μήκος παραθύρου για όλες τις δραστηριότητες. Η Εικόνα 14 δείχνει δεδομένα τριαξονικού επιταχυνσιόμετρου χρησιμοποιώντας χρονικό παράθυρο 2,54 δευτερολέπτων. Τέλος, κατά τη διαδικασία κατακερματισμού, πρέπει να οριστεί και το ποσοστό επικάλυψης του χρονικού παραθύρου (δηλ., πόσες τιμές ενός δείγματος δεδομένων είναι ίδιες με τις προηγούμενες).



Εικόνα 14: Γραφική παράσταση δεδομένων τριαξονικού επιταχυνσιόμετρου χρησιμοποιώντας χρονικό παράθυρο ίσο με 2,54 δευτερόλεπτα.

#### 2.5.4 Εξαγωγή Χαρακτηριστικών

Η ταξινόμηση των δεδομένων σήματος χωρίς τη σωστή επιλογή χαρακτηριστικών οδηγεί σε πολλές ανακριβείς ταξινομήσεις. Ως αποτέλεσμα, εφαρμόστηκε ένα ευρύ φάσμα αλγοριθμικών προσεγγίσεων με την πάροδο των ετών σε μία προσπάθεια εξάπλωσης των βασικών χαρακτηριστικών σήματος που επιτρέπουν στους προηγμένους αλγόριθμους μηχανικής μάθησης να ανακαλύψουν χρήσιμες πληροφορίες πλαισίου [98]. Αυτές οι προσεγγίσεις μετατρέπουν ή μετασχηματίζουν τα σήματα εισόδου σε διαφορετικές τομείς αντιπροσώπευσης. Σε κάθε τομέα χρησιμοποιούνται συγκεκριμένες μέθοδοι για την αφαίρεση δεδομένων ακατέργαστου σήματος και την παροχή νέων παραστάσεων δεδομένων που καθιστούν δυνατή την εφαρμογή αλγορίθμων υψηλότερου επιπέδου για την επίγνωση πλαισίου. Σύμφωνα με την [98] οι τεχνικές επεξεργασίας σημάτων ταξινομούνται σε τρεις ευρείς τομείς:

- Τομέας χρόνου (π.χ., μέσος όρος, διακύμανση, τυπική απόκλιση, ελάχιστη τιμή, μέγιστη τιμή, εύρος τιμών, συσχέτιση).
- Συχνότητα τομέα (π.χ., μετασχηματισμός Fourier, μετασχηματισμός κυματισμού).
- Διακριτά πεδία αναπαράστασης (π.χ., αποστάσεις σημείων με βάση την Ευκλείδεια απόσταση).

### 2.5.5 Ταξινόμηση

Η εφαρμογή αλγορίθμων μηχανικής μάθησης για την ταξινόμηση του σήματος είναι το τελευταίο βήμα της επεξεργασίας των σημάτων και χρησιμοποιεί ως είσοδο τα χαρακτηριστικά που εξάγονται από τον άνθρωπο. Συνήθεις αλγόριθμοι μηχανικής μάθησης, όπως ο αλγόριθμος  $k$  πλησιέστερου γείτονα ( $k$ -NN), οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines; SVM), τα Δέντρα Αποφάσεων (Decision Trees; DT), τα Τυχαία Δάση (Random Forests; RF) και τα Νευρωνικά Δίκτυα (Neural Networks), χρησιμοποιούνται για την κατάρτιση ενός μοντέλου που ταιριάζει καλύτερα στα δεδομένα της εκπαίδευσης (χωρίς να υπάρχει υπερπροσαρμογή του μοντέλου σε αυτά) για την αναγνώριση της σωματικής δραστηριότητας του χρήστη.

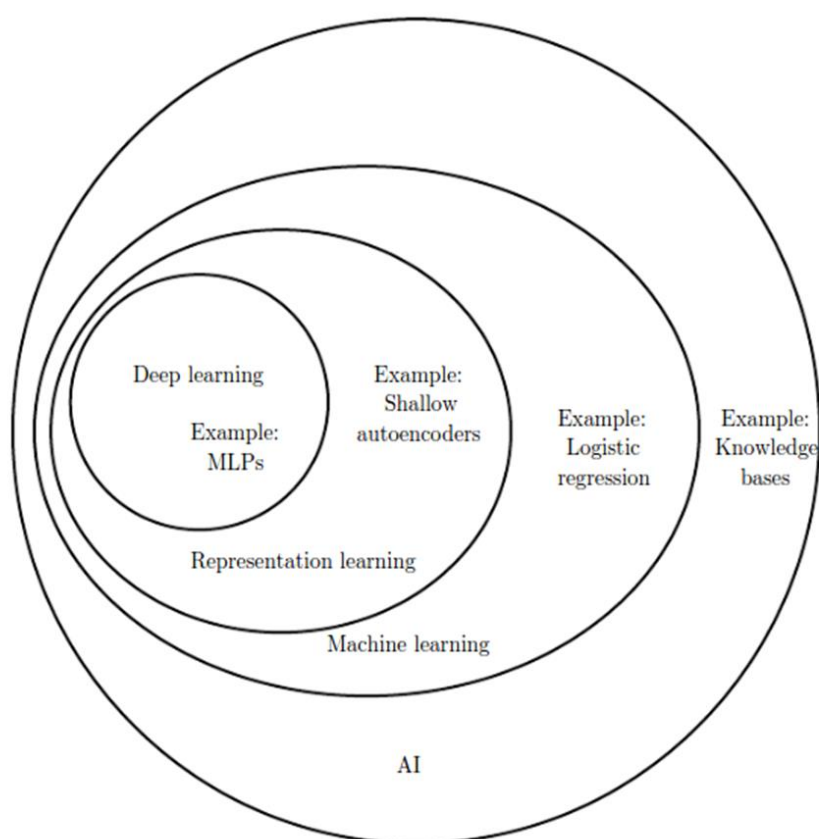
## 2.6 Βαθιά Μάθηση

Το κύριο μειονέκτημα της συνήθους διαδικασίας επεξεργασίας σήματος είναι το βήμα εξαγωγής χαρακτηριστικών. Όσο εξελιγμένος και να είναι ο αλγόριθμος ταξινόμησης, η ακρίβειά του εξαρτάται πολύ από τα επιλεγμένα χαρακτηριστικά. Επιπλέον, η εξαγωγή χαρακτηριστικών είναι χρονοβόρα και τις περισσότερες φορές τα επιλεγμένα χαρακτηριστικά δεν καταφέρνουν να εκφράσουν τις πολύπλοκες, μη γραμμικές σχέσεις/μοτίβα μεταξύ των δεδομένων των αισθητήρων. Ως αποτέλεσμα, πρόσφατα εισήχθησαν νέες τεχνικές αυτόματης εξαγωγής χαρακτηριστικών.

Η Βαθιά Μάθηση (Deep Learning; DL) είναι ένας κλάδος της Μηχανικής Μάθησης και, πιο συγκεκριμένα, είναι ένας όρος που δηλώνει ένα Βαθύ Νευρωνικό Δίκτυο. Ο λόγος που ονομάζεται βαθιά έγκειται στο γεγονός ότι αποτελείται από πολλά κρυφά επίπεδα σε αντίθεση με τα συμβατικά Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks; ANNs). Η Εικόνα 15 δείχνει τις ενσωματωμένες σχέσεις μεταξύ της Βαθιάς Μάθησης, της Μάθησης Χαρακτηριστικών (Representation Learning), της Μηχανικής Μάθησης και της Τεχνητής Νοημοσύνης.

Η Βαθιά Μάθηση είναι ένα πεδίο της Μάθησης Χαρακτηριστικών, η οποία αποτελεί ένα σύνολο τεχνικών που βοηθούν ένα σύστημα να εξαγάγει αυτόματα χαρακτηριστικά (δηλ., να ανακαλύψει πρότυπα) από ακατέργαστα δεδομένα και να τα «ταΐσει» στη συνέχεια σε έναν αλγόριθμο ταξινόμησης. Η διαφοροποίηση μεταξύ της Μάθησης Χαρακτηριστικών και της Βαθιάς Μάθησης οφείλεται στο ότι οι

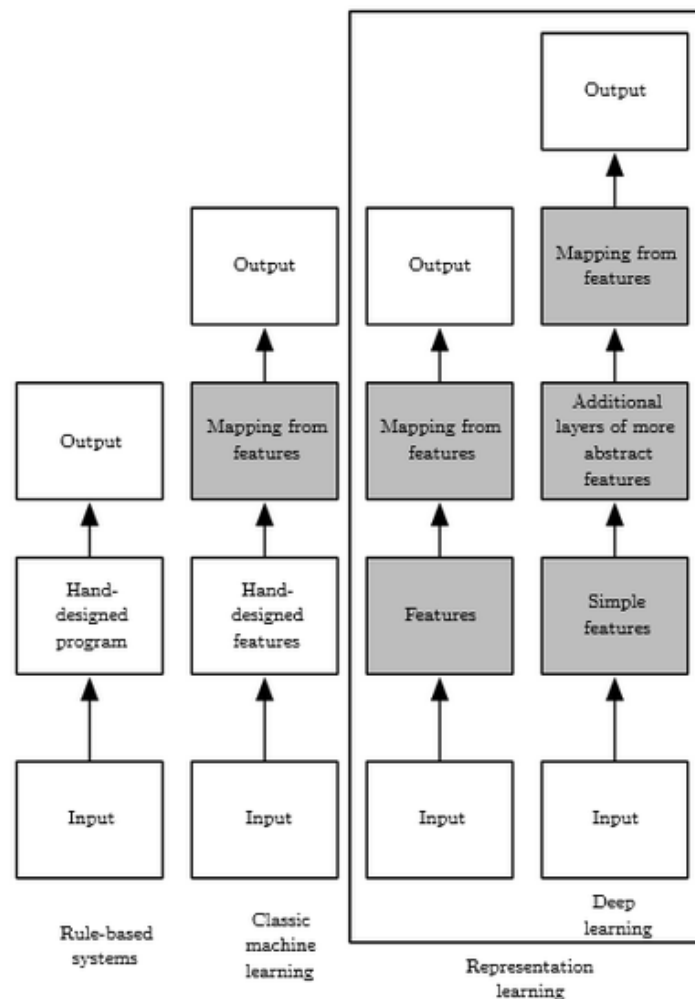
αλγόριθμοι Βαθιάς Μάθησης αποτελούνται από πολλαπλά επίπεδα ικανά να εξάγουν και χαρακτηριστικά υψηλού επιπέδου (Εικόνα 16), όπως είναι η αναγνώριση των ακμών των αντικειμένων μίας εικόνας. Ωστόσο, ο πιο συνηθισμένος τρόπος εξαγωγής χαρακτηριστικών γίνεται από εμπειρογνώμονες (Εικόνα 16), και για αυτό η Μάθηση Χαρακτηριστικών θεωρείται ότι είναι ένα υποσύνολο της Μηχανικής Μάθησης. Η Μηχανική Μάθηση είναι κλάδος της Τεχνητής Νοημοσύνης, καθώς αποτελεί ένα σύνολο αλγορίθμων που τροφοδοτούνται από τα επεξεργασμένα δεδομένα και κάνουν προβλέψεις για αυτά. Αυτοί οι αλγόριθμοι καταφέρνουν να έχουν καλύτερα αποτελέσματα από τα συστήματα που είναι βασισμένα σε κανόνες, όταν τα πρότυπα που πρέπει να ανακαλυφθούν είναι πολύ περίπλοκα και δεν μπορούν να εκφραστούν με τη χρήση κανόνων.



**Εικόνα 15:** Venn διάγραμμα που παρουσιάζει ότι η Βαθιά Μάθηση είναι μία ερευνητική περιοχή της Μάθησης Χαρακτηριστικών, η οποία είναι ένα υποσύνολο της Μηχανικής Μάθησης, η οποία, και αυτή με τη σειρά της είναι ένα πεδίο της Τεχνητής Νοημοσύνης [99].

Παρόλο που η Βαθιά Μάθηση έχει μακρά ιστορία, είχε παρουσιαστεί παλαιότερα υπό τον όρο Αυτοδίδακτη Μάθηση (Self-Taught Learning) [100], κέρδισε πολύ μεγάλη προσοχή τα τελευταία πέντε χρόνια. Οι πρώτες προσπάθειες για την προσθήκη περισσότερων κρυφών επιπέδων σε ένα ANN έγιναν στα μέσα της δεκαετίας του '80, ωστόσο τα αποτελέσματα που επιτεύχθηκαν δεν ήταν ικανοποιητικά λόγω του γεγονότος ότι:

- τα σύνολα δεδομένων που ήταν επισημασμένα ήταν πολύ μικρά σε μέγεθος εκείνες τις ημέρες,
- οι δυνατότητες επεξεργασίας των υπολογιστών ήταν πολύ περιορισμένες,
- η αρχικοποίηση των βαρών δεν γινόταν σωστά, και
- οι μη γραμμικές συναρτήσεις που εφαρμόζονταν, όπως είναι η σιγμοειδής συνάρτηση (sigmoid) και η συνάρτηση εφαπτομένης (tanh), ήταν λανθασμένες.



**Εικόνα 16:** Διαγράμματα ροής που παρουσιάζουν πως οι διάφορες διαδικασίες ενός συστήματος AI σχετίζονται μεταξύ τους στο πλαίσιο των διαφορετικών επιστημονικών κλάδων της AI. Τα σκιασμένα κουτιά υποδηλώνουν συστατικά στοιχεία που έχουν τη δυνατότητα να μαθαίνουν, από μόνα τους, κάνοντας χρήση των δεδομένων εισόδου [99].

Τα τελευταία δύο γεγονότα οδήγησαν στο πρόβλημα της εξαφάνισης της κλίσης της παραγώγου (vanishing gradient problem), κατά το οποίο το σφάλμα ταξινόμησης που χρησιμοποιείται για την οπίσθια διάδοση (back-propagation) «εξαφανίζεται» (αποκτά σχεδόν μηδενικές τιμές) όταν μεταδίδεται μέσω του δικτύου. Ως

αποτέλεσμα, τα βάρη των πρώτων στρωμάτων του ANN δεν ενημερώνονται στο βαθμό που θα έπρεπε.

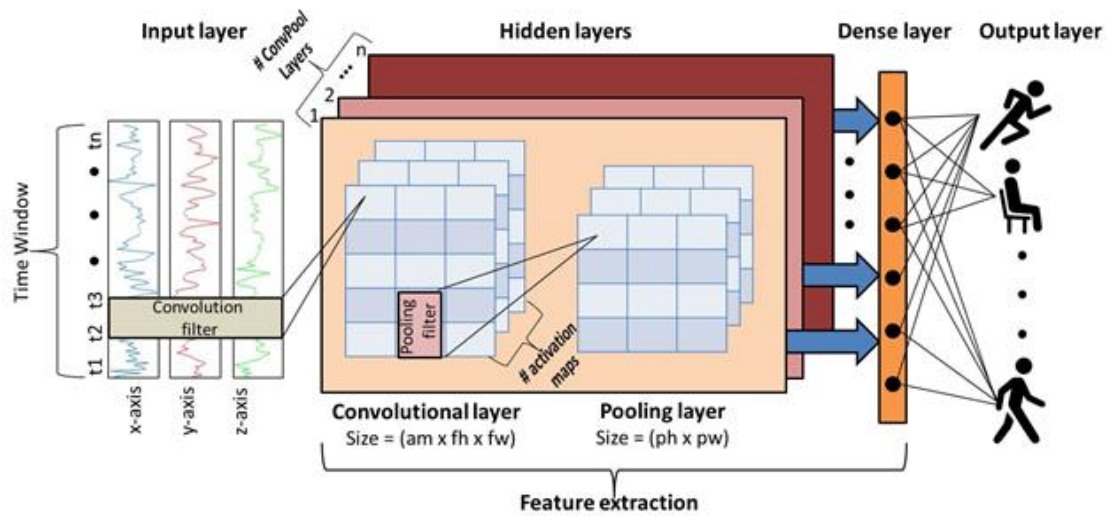
## 2.7 Συνελικτικά Νευρωνικά Δίκτυα

Ο πιο συνηθισμένος αλγόριθμος DL είναι τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks; CNNs). Ο Yann LeCun εισήγαγε το LeNet, το οποίο ήταν το πρώτο CNN στην [101]. Με βάση τη μαθηματική λειτουργία της συνέλιξης, το μοντέλο LeNet κατάφερε να ξεπεράσει τους άλλους αλγόριθμους ταξινόμησης αναγνωρίζοντας τα χειρόγραφα ψηφία [102]. Ωστόσο, τα ConvNets (Convolutional Networks) προσέλκυσαν την προσοχή του επιστημονικού κοινού σχεδόν 15 χρόνια αργότερα, όταν το βαθύ ConvNet του A. Krizhevsky και λοιπών [11], που ονομάστηκε AlexNet, κατάφερε να ξεπεράσει το δεύτερο στο διαγωνισμό αλγόριθμο κατά σχεδόν 5% χρησιμοποιώντας το σύνολο δεδομένων ImageNet [103].

Στα μαθηματικά, η συνέλιξη είναι ένας τρόπος συνδυασμού δύο συναρτήσεων για να σχηματισθεί μία τρίτη συνάρτηση. Στα CNNs, οι δύο συνδυασμένες συναρτήσεις είναι η είσοδος (δηλ., τα εικονοστοιχεία μίας εικόνας) και τα φίλτρα (βάρη), τα οποία πρέπει να εκπαιδευθούν για να ανακαλύψουν τις πιο αποτελεσματικές τιμές ώστε να εξαγάγουν τα απαραίτητα χαρακτηριστικά (π.χ., τις άκρες των αντικειμένων σε μία εικόνα). Η συνάρτηση εξόδου ονομάζεται χάρτης χαρακτηριστικών (feature map), που είναι προϊόν πολλών παράλληλων συνελίξεων μεταξύ της εισόδου και διαφορετικών φίλτρων. Στη συνέχεια, όπως συμβαίνει με τα κοινά Νευρωνικά Δίκτυα, κάθε χάρτης χαρακτηριστικών περνάει από μία μη γραμμική συνάρτηση ενεργοποίησης και γίνεται επεξεργασία του αποτελέσματος από μία συνάρτηση υποδειγματοληψίας (pooling), η οποία αντικαθιστά την έξοδο του δικτύου σε μία συγκεκριμένη περιοχή εφαρμόζοντας μία στατιστική συνάρτηση (π.χ., μέγιστη ή μέση τιμή) των γειτονικών εξόδων [104]. Η τοποθέτηση πολλών επιπέδων αυτών των λειτουργιών (ConvPool) οδηγεί στη δημιουργία βαθιών αρχιτεκτονικών CNN, γνωστών και ως DL.

Τα Βαθιά CNNs εκμεταλλεύονται τη σημασιολογία και την τοπολογία πλέγματος των δεδομένων εισόδου, σε αντίθεση με τα Νευρωνικά Δίκτυα. Συγκεκριμένα, τα στρώματα ενός CNN έχουν νευρώνες οργανωμένους σε 3 διαστάσεις: πλάτος, ύψος, βάθος. Για παράδειγμα, στην επεξεργασία εικόνας, οι εικόνες εισόδου έχουν τις εξής διαστάσεις: α) πλάτος εικόνας, β) ύψος εικόνας και γ) βάθος εικόνας (δηλ., μία εικόνα σε πρότυπο χρώματος RGB έχει βάθος 3, ενώ στην κλίμακα του γκρι έχει 1). Επιπλέον, οι νευρώνες σε ένα επίπεδο συνδέονται με μία μικρή περιοχή του προηγούμενου στρώματος (αραιή συνδεσιμότητα), αντί με όλους τους προηγούμενους νευρώνες (πυκνή συνδεσιμότητα). Ως αποτέλεσμα, η μνήμη και οι υπολογιστικές απαιτήσεις του αλγόριθμου μειώνονται. Η Εικόνα 17 παρουσιάζει ένα μοντέλο Βαθιού Συνελικτικού Νευρωνικού Δικτύου για ΑΑΔ χρησιμοποιώντας δεδομένα τριαξονικού

επιταχυνσιόμετρο. Τα επίπεδα και οι συναρτήσεις που το αποτελούν περιγράφονται στις παρακάτω ενότητες.



Εικόνα 17: Αρχιτεκτονική Βαθούς Συνελικτικού Νευρωνικού Δικτύου για αναγνώριση ανθρώπινης δραστηριότητας χρησιμοποιώντας δεδομένα τριαξονικού επιταχυνσιόμετρο [14].

### 2.7.1 Επίπεδο Συνέλιξης

Η λειτουργία συνέλιξης καταφέρνει να αποκτήσει μία λιγότερο θορυβώδη εκτίμηση των μετρήσεων ενός αισθητήρα, εκτιμώντας τη μέση τιμή τους. Λόγω του γεγονότος ότι μερικές μετρήσεις είναι πιο συναφείς (π.χ., κάποιες παρουσιάζουν υψηλές τιμές θορύβου) και ως αποτέλεσμα, θα πρέπει να συνεισφέρουν περισσότερο στο μέσο όρο, οι μετρήσεις των αισθητήρων συνυπάρχουν με μία συνάρτηση βαρύτητας  $w$  [104]. Συνεπώς, στην περίπτωση της ΑΑΔ, η είσοδος, η οποία είναι το προεπεξεργασμένο και κατακερματισμένο σήμα κίνησης, συνδυάζεται με τα φίλτρα (βάρη), τα οποία εκπαιδεύονται για να ανακαλύψουν τα καταλληλότερα μοτίβα (π.χ., κορυφές στο σήμα). Η έξοδος της συνέλιξης ονομάζεται *χάρτης ενεργοποίησης*, ο οποίος είναι προϊόν πολλών παράλληλων συνελίξεων του σήματος με τα φίλτρα.

Το στοιχείο  $i$  μίας διακεκριμένης συνέλιξης 1D μεταξύ της συστοιχίας εισόδου  $x$  και του μονοδιάστατου φίλτρου  $w$  ισούται με:

$$c_i^{l,q} = b^{l,q} + \sum_{d=1}^D w_d^{l,q} x_{i+d-1}^{l-1,q} \quad (2.1)$$

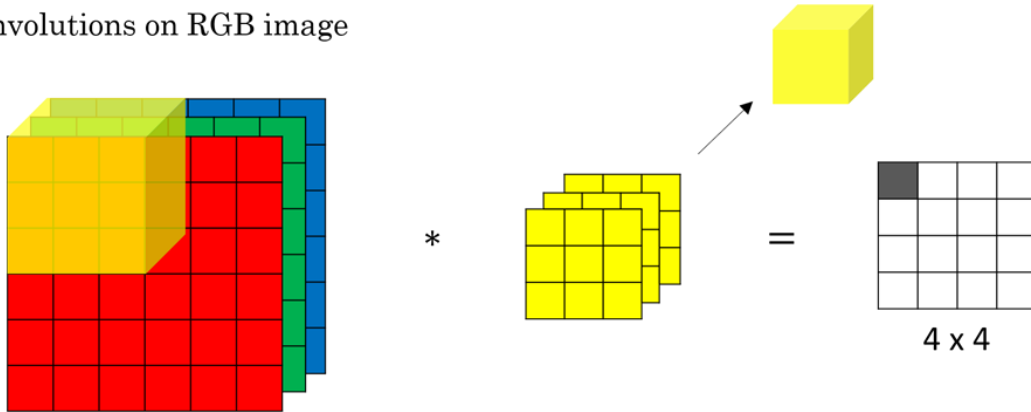
όπου  $l$  είναι ο δείκτης επιπέδου,  $q$  είναι ο δείκτης του χάρτη ενεργοποίησης,  $D$  είναι το συνολικό πλάτος του φίλτρου  $w$ , και  $b$  είναι ο όρος προκατάληψης (bias term). Ενώ

το στοιχείο  $i$  του προϊόντος μίας διακριτής δισδιάστατης συνέλιξης (Εικόνα 18) μεταξύ της εισόδου  $x$  και του δισδιάστατου φίλτρου  $w$  ισούται με:

$$c_{i,j}^{l,q} = b^{l,q} + \sum_{h=1}^H \sum_{d=1}^D w_{d,h}^{l,q} x_{i+d-1,i+h-1}^{l-1,q} \quad (2.2)$$

όπου το  $h$  είναι το συνολικό ύψος του φίλτρου  $w$ .

Convolutions on RGB image



Εικόνα 18: Δυσδιάστατη συνέλιξη σε μία RGB εικόνα [105].

### 2.7.2 Ενεργοποίηση ReLu

Μετά από μία συνέλιξη, όπως συμβαίνει και με τα συνηθισμένα ANN, κάθε χάρτης χαρακτηριστικών διατρέχεται από μία μη γραμμική συνάρτηση ενεργοποίησης. Χρησιμοποιείται στη συγκεκριμένη αρχιτεκτονική η συνάρτηση ενεργοποίησης ReLu (Rectified Linear unit) [11], η οποία περιορίζει όλες τις τιμές στο μηδέν και περιγράφεται ως εξής:

$$a = \max(0, c) \quad (2.3)$$

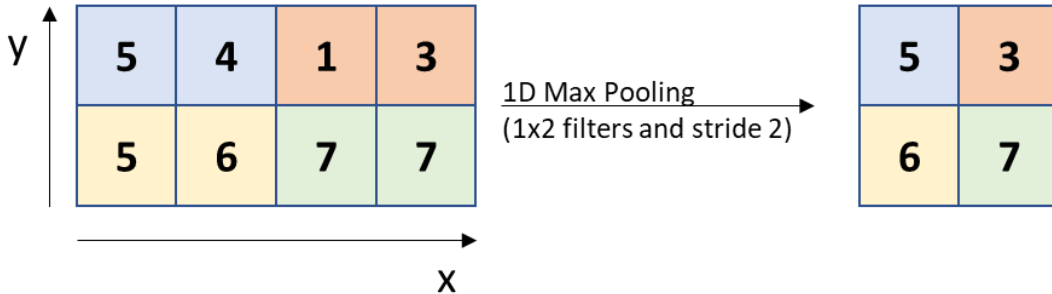
### 2.7.3 Επίπεδο υποδειγματοληψίας μέγιστης τιμής

Το αποτέλεσμα της συνάρτησης ενεργοποίησης το επεξεργάζεται μία συνάρτηση υποδειγματοληψίας [106], η οποία αντικαθιστά την έξοδο του δικτύου σε μία συγκεκριμένη περιοχή με μία συνοπτική στατιστική μεταξύ των γειτονικών εξόδων [104]. Επιπλέον, η λειτουργία υποδειγματοληψίας μειώνει το χωρικό μέγεθος της εισόδου, τον αριθμό των παραμέτρων και, συνεπώς, τον αριθμό των υπολογισμών που απαιτούνται. Η μονοδιάστατη συνάρτηση υποδειγματοληψίας μέγιστης τιμής (Εικόνα 19) επιλέγει τη μέγιστη τιμή της συστοιχίας εισόδου:

$$p_i^{l,q} = \max(a_{i \times s+z}^{l,q}) \quad (2.4)$$



όπου  $z$  είναι το μέγεθος υποδειγματοληψίας και το  $s$  δηλώνει το μέγεθος του βήματος (δηλ., κατά πόσες θέσεις κινείται ο αλγόριθμος μετά την εκτέλεση της διαδικασίας υποδειγματοληψίας).



Εικόνα 19: Μονοδιάστατη (μόνο ως προς τον άξονα X) υποδειγματοληψία μέγιστης τιμής.

### 2.7.4 Επίπεδο ολικής υποδειγματοληψίας μέσης τιμής

Το επίπεδο ολικής υποδειγματοληψίας μέσης τιμής (Global average pooling layer) είναι παρόμοιο με το πλήρως συνδεδεμένο επίπεδο (Dense Layer), καθώς και οι δύο εκτελούν γραμμικούς μετασχηματισμούς των χαρτών ενεργοποίησης. Η συνάρτηση είναι παρόμοια με την υποδειγματοληψία μέγιστης τιμής, αλλά επιστρέφει τη μέση τιμή ολόκληρου του χάρτη ενεργοποίησης, αντί μίας περιοχής αυτού. Σύμφωνα με την [107], η ολική υποδειγματοληψία μέσης τιμής λειτουργεί ως εξομάλυνση και σε σύγκριση με ένα πλήρως συνδεδεμένο στρώμα, επιτυγχάνει μεγαλύτερη ακρίβεια. Η αποτελεσματικότητά του βασίζεται στο γεγονός ότι επιβάλλει την αντιστοιχία μεταξύ χαρτών ενεργοποίησης και κατηγοριών. Η ολική υποδειγματοληψία μέσης τιμής ισούται με:

$$p_q^l = \frac{1}{N} \sum_{i=1}^N (a_i^{l,q}) \quad (2.5)$$

### 2.7.5 Συνάρτηση ενεργοποίησης Softmax

Το τελευταίο στρώμα ενός κοινού ANN ή ενός CNN περιλαμβάνει τη συνάρτηση ενεργοποίησης Softmax. Η συνάρτηση Softmax χρησιμοποιείται σε μεθόδους πολλαπλής ταξινόμησης, δεδομένου ότι έχοντας μία είσοδο  $p$ , βάρη  $w$  και προκατάληψη  $b$ , εκτιμά την πιθανότητα αυτή η είσοδος να ανήκει σε μία συγκεκριμένη κλάση  $k$ :

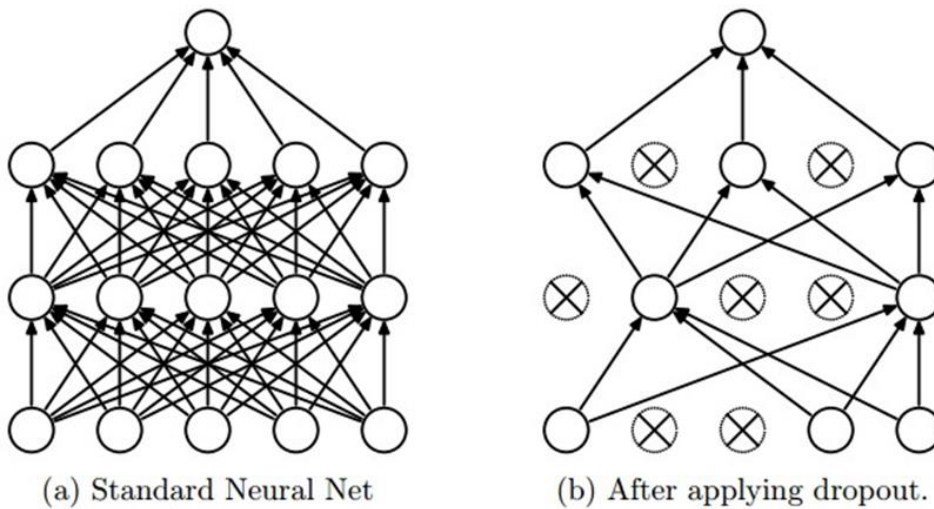
$$P(y = k|p; w; b) = \frac{\exp(b^L + w^L p^{L-1})}{\sum_{k=1}^K \exp(b^L + w_k^L p^{L-1})} \quad (2.6)$$

όπου  $K, L$  είναι ο συνολικός αριθμός κατηγοριών ΑΑΔ και τα επίπεδα αντίστοιχα. Η κλάση που πέτυχε την υψηλότερη πιθανότητα θεωρείται ότι είναι το αποτέλεσμα ολόκληρου του μοντέλου CNN:

$$Y = \operatorname{argmax}(P(y = k|p; w; b)) \quad (2.7)$$

### 2.7.6 Επίπεδο απόρριψης συνδέσεων dropout

Η τεχνική απόρριψης συνδέσεων που εισάγεται στην [108], αντιπροσωπεύει την πιθανότητα  $p$  για τη διατήρηση ενός ενεργού νευρώνα ή το μηδενισμό του. Κατά τη διάρκεια της απόρριψης dropout, το δίκτυο επιτυγχάνει να βρει πιο γενικά/αφαιρετικά μοτίβα, τα οποία δεν είναι επιρρεπή στην υπερπροσαρμογή στα δεδομένα εκπαίδευσης (Εικόνα 20).



Εικόνα 20: Απεικόνιση απόρριψης συνδέσεων στο δεξί νευρωνικό δίκτυο (b), σε αντίθεση με το αριστερό (a) [108].

### 2.7.7 Συνάρτηση Κόστους

Η λειτουργία κόστους υπολογίζει τη διαφορά μεταξύ της επισήμανσης (επιθυμητής κλάσης) και της πρόβλεψης και χρησιμοποιείται για την τροφοδοσία του αλγόριθμου βελτιστοποίησης. Η ελαχιστοποίηση της συνάρτησης κόστους σημαίνει ότι οι προβλέψεις του αλγόριθμου βελτιώνονται. Για να αντιπροσωπευθεί αυτή η απόκλιση χρησιμοποιείται η αρνητική πιθανότητα λογαρίθμου, η οποία ισοδυναμεί με τη μεγιστοποίηση της πιθανότητας του συνόλου δεδομένων  $D$  όταν το μοντέλο του CNN παραμετροποιείται από τις παραμέτρους  $\theta$  (φίλτρο  $w$  και προκατάληψη  $b$ ). Συνεπώς, η λειτουργία κόστους ορίζεται ως:

$$L(\theta) = - \sum_{i=0}^D \log(P(Y = y^i | x^i, \theta)) \quad (2.8)$$

## 2.8 Αλγόριθμοι Βαθιάς Μάθησης για Αναγνώριση Ανθρώπινης Δραστηριότητας

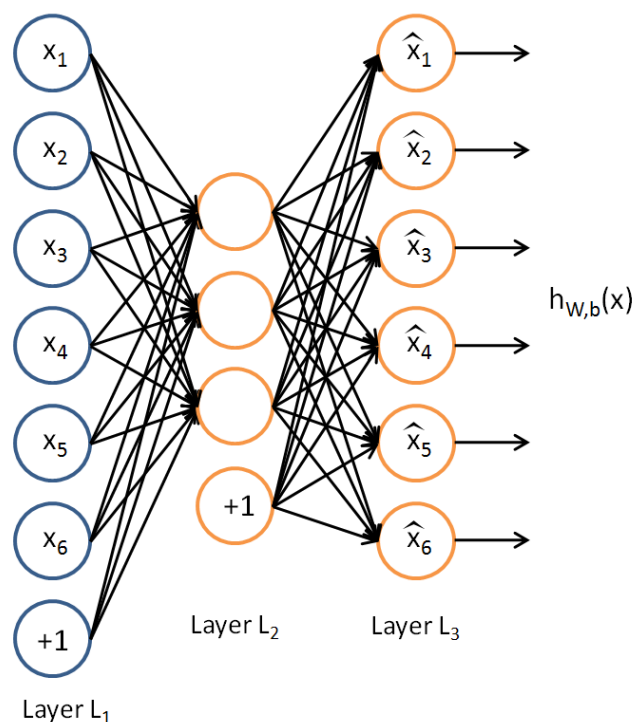
Οι πρόσφατες εξελίξεις στον τομέα της μηχανικής μάθησης επηρέασαν, ασφαλώς, και τις μεθόδους ΑΑΔ. Έτσι, οι προσεγγίσεις της Βαθιάς Μάθησης για την ΑΑΔ θεωρούνται ως τεχνολογίες αιχμής, και μπορούν να ταξινομηθούν ως εξής: α) Αυτοκωδικοποιητές (Autoencoders), β) Συνελικτικά Νευρωνικά Δίκτυα, γ) Βαθιά Μάθηση σε φασματογραφήματα και δ) Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks; RNNs).

### 2.8.1 Αυτοκωδικοποιητές

Οι Αυτοκωδικοποιητές είναι μία συγκεκριμένη κατηγορία των Τεχνητών Νευρωνικών Δικτύων, που βασίζεται στη Μη-επιτηρούμενη Μάθηση (Unsupervised Learning), δηλαδή Μηχανική Μάθηση βασισμένη σε μη χαρακτηρισμένα δεδομένα. Η διάταξη του συγκεκριμένου νευρωνικού αποτελείται από μία είσοδο, και ένα κρυφό επίπεδο που προσπαθεί να παράγει ως έξοδο τις τιμές εισόδου. Στην Εικόνα 21 παρουσιάζεται ένας Αυτοκωδικοποιητής, με το σύνολο των  $x_i$  να αποτελεί την είσοδο και το  $\hat{x}_i$  την έξοδο. Το νευρωνικό μοντέλο κατά την εκπαίδευσή του προσπαθεί να μηδενίσει τη μέση απόκλιση,  $\sum_i^n (x_i - \hat{x}_i)$ . Πολλές στοίβες από Αυτοκωδικοποιητές σχηματίζουν ένα Βαθύ Νευρωνικό Δίκτυο.

Στην [109] χρησιμοποιήθηκε μία τεχνική Autoencoder για την εξαγωγή βασικών χαρακτηριστικών για ΑΑΔ. Συγκεκριμένα, οι συγγραφείς χρησιμοποίησαν τις Περιορισμένες Μηχανές Boltzmann (Restricted Boltzmann Machines; RBM) [110], οι οποίες αποτελούν μία συγκεκριμένη μορφή γραμμικών λογαριθμικών Τυχαίων Μαρκοβιανών Πεδίων (Markov Random Fields), δηλαδή ενός μη κατευθυνόμενου πιθανολογικού γραφικού μοντέλου και έχουν εφαρμοσθεί επιτυχώς για τη μείωση των διαστάσεων των δεδομένων στην υπολογιστική όραση [111]. Μία άλλη προσέγγιση Autoencoder προτάθηκε στη [112], όπου ο C. Vollmer και λοιποί χρησιμοποιούν έναν Αραιά Συνδεδεμένο Αυτοκωδικοποιητή (Sparse Autoencoder) [113] για την εξαγωγή χαρακτηριστικών.

Ωστόσο, θα πρέπει να σημειωθεί ότι οι Αυτοκωδικοποιητές είναι πλήρως συνδεδεμένα μοντέλα και ως εκ τούτου δεν καταφέρνουν να καταγράψουν με επιτυχία τις τοπικές εξαρτήσεις της χρονοσειράς δεδομένων από τους αισθητήρες [114][115]. Κάτι που ισχύει και στην περίπτωση των Sparse Autoencoders, καθώς οι συνδέσεις των ενδιάμεσων νευρώνων με τα δεδομένα εισόδου που επιλέγονται κατά την εκπαίδευση του μοντέλου είναι τυχαίες.



Εικόνα 21: Απεικόνιση Αυτοκωδικοποιητή [116].

### 2.8.2 Συνελκτικά Νευρωνικά Δίκτυα για Αναγνώριση Ανθρώπινης Δραστηριότητας

Η πρώτη προσέγγιση CCN στην ΑΑΔ παρουσιάζεται στην [117]. Οι συγγραφείς χρησιμοποίησαν ως είσοδο τα μονοδιάστατα (1D) σήματα κίνησης, σε αντίθεση με την ανάλυση εικόνας που χρησιμοποιεί μία συστοιχία ρικελ δύο διαστάσεων (2D). Με αυτόν τον τρόπο, τα σήματα στοιβάζονται σε κανάλια (στοίβαξη βάσει καναλιών), δηλαδή αντιστοιχεί ένα κανάλι σε κάθε διαφορετικό σήμα. Για παράδειγμα, ένας αισθητήρας τριαξονικού επιταχυνσιόμετρου παράγει 3 κανάλια (άξονες X, Y, Z), παρόμοια με τις έγχρωμες εικόνες (κόκκινο, μπλε, πράσινο κανάλια). Ως αποτέλεσμα, η λειτουργία συνέλιξης εφαρμόζεται σε κάθε σήμα ξεχωριστά. Μετά από δύο διαδοχικές συναρτήσεις συνέλιξης και υποδειγματοληψίας (ConvPool) (Ενότητα 2.7.3), μετατρέπουν την έξοδό τους από έναν τρισδιάστατο τανυστή σε ένα διάνυσμα και εφαρμόζουν ένα πλήρως συνδεδεμένο νευρωνικό επίπεδο (Dense Layer), για να κάνουν την ταξινόμηση.

Ομοίως, στην [115] προτείνουν την ίδια αρχιτεκτονική, αλλά χρησιμοποιούν ένα πιο «ρηχό» CNN (μόνο μία συνάρτηση ConvPool) και έχουν σαν είσοδο τις χρονοσειρές της τριαξονικής επιτάχυνσης. Τα αποτελέσματα της έρευνας έδειξαν ότι η χρήση ενός μοντέλου CNN ξεπερνά σε ακρίβεια τους Αυτοκωδικοποιητές. Επιπλέον, ο Ropao και λοιποί [118][119][120], αλλά και ο J. B. Yang και λοιποί [121] προτείνουν μία βαθύτερη προσέγγιση με χρήση CNN (3 επίπεδα ConvPool) στην ΑΑΔ, χρησιμοποιώντας ένα σύνολο δεδομένων κίνησης, αποτελούμενο από σήματα

τριαξονικού γυροσκοπίου και επιταχυνσιόμετρου. Επιπλέον, η εισαγωγή πρόσθετων πληροφοριών ως εισροών, με την προσθήκη των τιμών του μετασχηματισμού Fourier στα αρχικά σήματα, κατάφερε να επιτύχει σχεδόν 1% υψηλότερη ακρίβεια [118]. Παρ'όλα αυτά, όλες οι προαναφερθείσες προσεγγίσεις που βασίζονται σε CNNs χρησιμοποιούν σαν τελικό κρυφό επίπεδο ένα πλήρως συνδεδεμένο νευρωνικό, και παρόμοια με τους Αυτοκωδικοποιητές δεν καταφέρνουν να εντοπίσουν κάποια τοπικά εξαρτώμενα από το χρόνο μοτίβα.

### 2.8.3 Βαθιά Μάθηση σε φασματογραφήματα

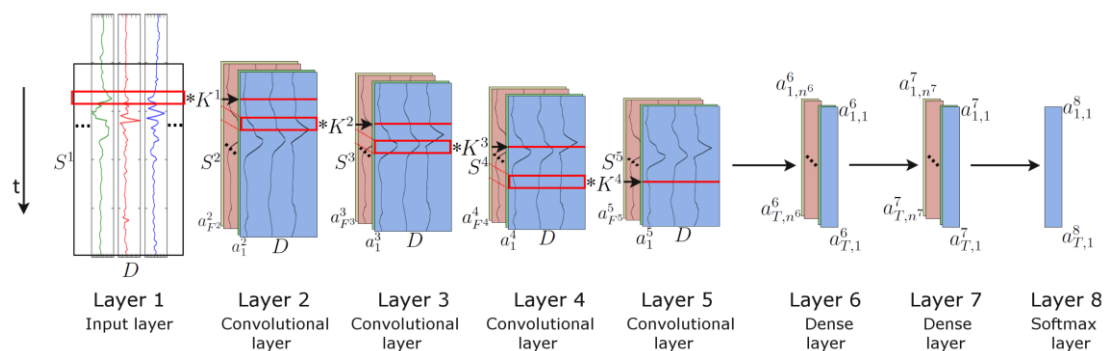
Μία ενδιαφέρουσα προσέγγιση η οποία εφαρμόζεται σε ακουστικά σήματα είναι αυτή της μετατροπής του σήματος του αισθητήρα εισόδου σε φασματογράφημα (δηλ., κατά τη διάρκεια του βήματος εξαγωγής χαρακτηριστικών), παρέχοντας μία αναπαράσταση του σήματος ως συνάρτηση της συχνότητας και του χρόνου. Στη συνέχεια, η εικόνα φασματογράφων τροφοδοτεί ένα Βαθύ Νευρωνικό Δίκτυο, όπως και στη διαδικασία ανάλυσης εικόνας.

Οι συγγραφείς της [122] υιοθετούν μία υβριδική προσέγγιση, η οποία βασίζεται σε ένα μοντέλο DL και στα Κρυφά Μαρκοβιανά Μοντέλα (Hidden Markov Models; HMM) για τη διαδοχική αναγνώριση της ανθρώπινης δραστηριότητας. Συγκεκριμένα, το σήμα του τριαξονικού επιταχυνσιόμετρου μεταφράζεται σε φασματογράφημα και στη συνέχεια εφαρμόζεται σε αυτό ένα RBM. Επιπλέον, ένα μη υποχρεωτικό βήμα HMM, το οποίο έχει ως είσοδο τις πιθανότητες εκπομπής από το RBM, εφαρμόζεται για τη μοντελοποίηση χρονικών μοτίβων, όσον αφορά τις ανθρώπινες δραστηριότητες. Μία πιο εξελιγμένη τεχνική υιοθετείται στην [123], όπου το CNN εφαρμόζεται σε μία *εικόνα δραστηριότητας (activity image)*. Σύμφωνα με τους συγγραφείς, τα ακατέργαστα σήματα του τριαξονικού επιταχυνσιόμετρου και του γυροσκοπίου στοιβάζονται κατά σειρά σε μία *εικόνα σήματος (signal image)*, με βάση έναν αλγόριθμο. Ακολούθως εφαρμόζεται ένας 2D διακριτός μετασχηματισμός Fourier (Discrete Fourier Transform; DFT) στην *εικόνα σήματος* και το μέγεθος του (magnitude) αντιπροσωπεύει την *εικόνα δραστηριότητας*. Με αυτόν τον τρόπο, οι ακολουθίες του σήματος εισόδου είναι δίπλα η μία στην άλλη, επιτρέποντας στο CNN να εξαγάγει κρυφές συσχετίσεις μεταξύ γειτονικών σημάτων. Ωστόσο, πρέπει να σημειωθεί ότι η μετατροπή των δεδομένων χρονοσειρών στον τομέα συχνοτήτων δεν είναι τόσο αποτελεσματική στην ΑΑΔ όσο στην ταξινόμηση του ήχου και τα στατιστικά χαρακτηριστικά του χρόνου έχουν αποδειχθεί πιο σημαντικά [98].

### 2.8.4 Επαναλαμβανόμενα Νευρωνικά Δίκτυα

Τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα [124] είναι μία οικογένεια νευρωνικών δικτύων για την επεξεργασία μίας ακολουθίας τιμών. Ως αποτέλεσμα, τα RNNs εφαρμόζονται ευρέως σε δεδομένα χρονοσειρών. Επιπλέον, επειδή στην

περίπτωση διαδοχικών δεδομένων, μία τιμή  $x_i$  εξαρτάται από ένα σύνολο προηγούμενων τιμών  $n \{x_{i-1}, x_{i-2}, \dots, x_{i-n}\}$  και από ένα σύνολο επόμενων τιμών  $n \{x_{i+1}, x_{i+2}, \dots, x_{i+n}\}$ , εφαρμόζεται ένας μηχανισμός που ονομάζεται Μακρο-Βραχυπρόθεσμη Μνήμη (Long Short-Term Memory; LSTM) [125] για την ενίσχυση της μνήμης του δικτύου.



**Εικόνα 22: Απεικόνιση της αρχιτεκτονικής του μοντέλου Convolutional LSTM σε σήματα αισθητήρων κίνησης [126].**

Μία υβριδική προσέγγιση που ονομάζεται Convolutional LSTM παρουσιάζεται στην [126]. Αυτό το δίκτυο αποτελείται από τέσσερις διαδοχικές συναρτήσεις συνέλιξης, όπου η έξοδος του τελευταίου τροφοδοτεί ένα επίπεδο LSTM. Στη συνέχεια, χρησιμοποιείται ένα τελικό επίπεδο LSTM για την πρόβλεψη της κλάσης της εκτελεσθείσας δραστηριότητας (Εικόνα 22). Αυτή η προσέγγιση RNN εφαρμόστηκε σε δύο σύνολα δεδομένων ΑΑΔ και κατόρθωσε να μοντελοποιήσει τις χρονικές εξαρτήσεις αποτελεσματικότερα από ένα συνηθισμένο ConvNet.

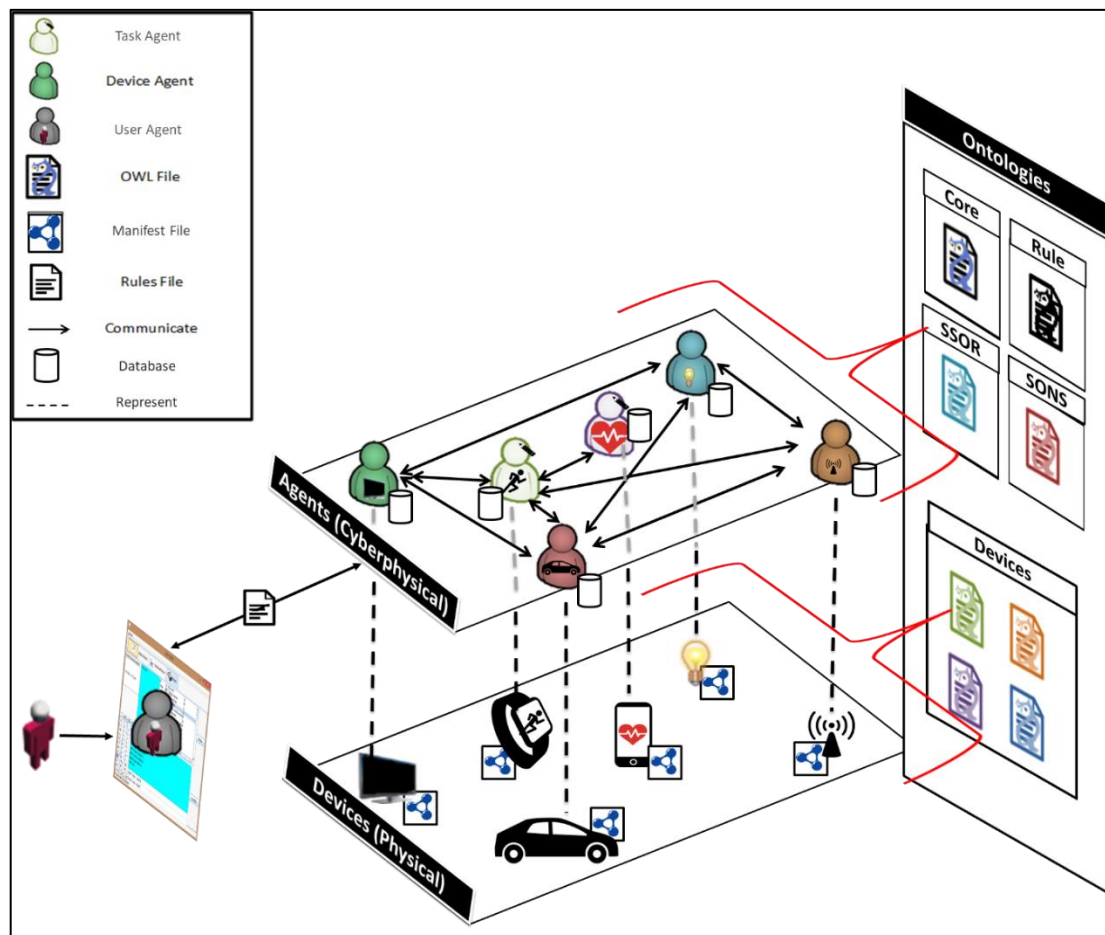
## 3 Αρχιτεκτονική πλατφόρμας και γνωσιακοί πράκτορες για το Κοινωνικό Διαδίκτυο των Πραγμάτων

### 3.1 Προτεινόμενη Αρχιτεκτονική για το Κοινωνικό Διαδίκτυο των Πραγμάτων

Η προτεινόμενη πλατφόρμα *SlIoT* βασίζεται σε ένα οικοσύστημα *SlIoT*, το οποίο κάνει χρήση σημασιολογικών κανόνων και τεχνικών Βαθιάς Μάθησης για τη διαχείριση των ενεργειών των έξυπνων αντικειμένων. Για να επιτευχθεί αυτό, η υλοποίηση της πλατφόρμας βασίζεται σε δύο επίπεδα: (α) Το Επίπεδο Συσκευής (*Device Layer*), το οποίο είναι υπεύθυνο για τη διαχείριση των συσκευών (οι οποίες μπορεί να μην έχουν απαραίτητα φυσική παρουσία, αλλά θα μπορούσαν να είναι εικονικές) και υλοποιεί τις απαραίτητες διεπαφές για τις συσκευές και (β) το Επίπεδο Πράκτορα (*Agent Layer*), το οποίο περιλαμβάνει τους πράκτορες λογισμικού που θα είναι συνδεδεμένοι με τις συσκευές, θα είναι υπεύθυνοι για την υλοποίηση της λογικής που συνδέεται με τη λειτουργία κάθε συσκευής και τις αντίστοιχες λειτουργίες επικοινωνίας (Ενότητα 3.2).

Η Εικόνα 23 παρουσιάζει την προτεινόμενη αρχιτεκτονική για το *SlIoT* οικοσύστημα. Κάθε στρώμα της αρχιτεκτονικής περιέχει τις δικές του οντότητες και για κάθε στρώμα χρησιμοποιούνται διαφορετικά οντολογικά σχήματα. Συγκεκριμένα, στο *Device Layer* χρησιμοποιείται η οντολογία για τη διαχείριση συσκευών, ενώ στο *Agent Layer* χρησιμοποιείται η κεντρική οντολογία, η οντολογία κανόνων, η οντολογία διαχείρισης κοινωνικών δεσμών, και η οντολογία διαχείρισης αναγκών και στόχων (Ενότητα 3.4.3). Το χαμηλότερο στρώμα συνδέεται με τις υλικές οντότητες, οι οποίες μπορεί να είναι είτε φυσικά συστήματα όπως συσκευές, ενεργοποιητές, αισθητήρες ή ακόμη και διεπαφές εργαλείων ή εφαρμογών που μπορούν να παράγουν ή να δέχονται καλά καθορισμένες τιμές και παραμέτρους που αντιστοιχούν σε φυσικά χαρακτηριστικά (π.χ., εφαρμογή υγείας). Το ανώτερο στρώμα αντιπροσωπεύει τη σύνδεση με τον κυβερνο-φυσικό κόσμο, επιτρέποντας στις οντότητες του στρώματος των συσκευών να επικοινωνούν και να λειτουργούν μέσω άμεσων εντολών των χρηστών (ιδιοκτητών) ή αυτόνομα, μέσω της εφαρμογής των κανόνων που τους αντιστοιχούν. Για να υποστηρίξει η πλατφόρμα τη διεπαφή ανθρώπου-μηχανής, το σχεδιασμό, τη δοκιμή και την εφαρμογή αυτών των κανόνων, χρησιμοποιείται ένα ειδικό επίπεδο, το οποίο λειτουργεί ως Πράκτορας Χρήστη (*User Agent*). Ο Πράκτορας Χρήστη χρησιμοποιείται για την αποστολή μηνυμάτων στους πράκτορες λογισμικού όταν συμβαίνει ένα γεγονός (δηλ., εισαγωγή νέου κανόνα) και καταγράφει την οντολογική αναπαράσταση του κανόνα σε ένα αρχείο γλώσσας *turtle*, στο οποίο έχει πρόσβαση μόνο ο πράκτορας της οντότητας που θα

λειτουργήσει ως δράστης (Ενότητα 3.4.4). Εάν αυτός ο νέος κανόνας δεν έρχεται σε σύγκρουση με κάποιον άλλο κανόνα, ο Πράκτορας Χρήστη το καταγράφει σε ένα αρχείο κειμένου που περιέχει όλους τους άλλους κανόνες και το στέλνει στον αντίστοιχο Device Agent μέσω ενός μηνύματος πληροφορίας.



Εικόνα 23: Αρχιτεκτονική του SIoT οικοσυστήματος.

### 1. Επίπεδο Συσκευής

Οι οντότητες του *Device Layer* είναι οι αισθητήρες, οι ενεργοποιητές, τα έξυπνα αντικείμενα ή ακόμη και οι εφαρμογές οι οποίες είναι εγκατεστημένες σε φορητές/κινητές συσκευές. Το επίπεδο της συσκευής απεικονίζει το φυσικό στρώμα της πλατφόρμας. Μία έξυπνη οντότητα έρχεται μαζί με την οντολογία του τομέα (*Domain Ontology*) που καθορίζεται από τον κατασκευαστή της. Όταν η οντότητα συνδεθεί με την πλατφόρμα, η οντολογία της αποθηκεύεται στο χώρο αποθήκευσης οντολογιών. Κάθε αρχείο OWL περιέχει σημαντικές υποδείξεις για μία συγκεκριμένη συσκευή, όπως τις διαθέσιμες καταστάσεις, τις εντολές που δέχεται, τις υπηρεσίες που προσφέρει και τους σημασιολογικούς κανόνες που έχουν γραφτεί από τον κατασκευαστή. Όλες αυτές οι οντολογίες τομέα αντιστοιχίζονται στην κεντρική οντολογία (*Core Ontology*).



## 2. Επίπεδο Πράκτορα

Το Επίπεδο Πράκτορα περιέχει γνωσιακούς πράκτορες, τα στοιχεία λογισμικού δηλαδή που καθορίζουν τη συμπεριφορά μίας συσκευής, μίας εφαρμογής ή του χρήστη. Κάθε έξυπνη οντότητα του Διαδικτύου των Πάντων αντιπροσωπεύεται από έναν Device/Task/User Agent, ο οποίος είναι υπεύθυνος για τον έλεγχό της. Είναι μία σχέση ενός προς ένα, όπου η οντότητα αναφέρει στον πράκτορά της αλλαγές στην κατάσταση της ή αλλαγή στην παράμετρο που παρακολουθεί (π.χ., θερμοκρασία). Όλοι οι άλλοι πράκτορες, οι οποίοι έχουν εγγραφεί σε ένα συγκεκριμένο θέμα της κατάστασής της οντότητας, ενημερώνονται για πιθανές αλλαγές μέσω του προτύπου Δημοσίευσης-Συνδρομής. Η προαναφερθείσα σύνδεση μεταξύ των πρακτόρων λογισμικού είναι συνέπεια των σημασιολογικών κανόνων που έχει εισάγει ο χρήστης, μέσω της κατάλληλης διεπαφής (Ενότητα 4.1), οι οποίοι καθορίζουν τους στόχους του πράκτορα. Όταν συμβαίνει μία αλλαγή σε μία παράμετρο, ο πράκτορας αναλύει τους σημασιολογικούς κανόνες που σχετίζονται με τη συγκεκριμένη μεταβλητή και με χρήση παραγωγικού συλλογισμού, στέλνει ένα αίτημα εντολής, αν είναι απαραίτητο, για να αλλάξει η κατάσταση του έξυπνου αντικειμένου.

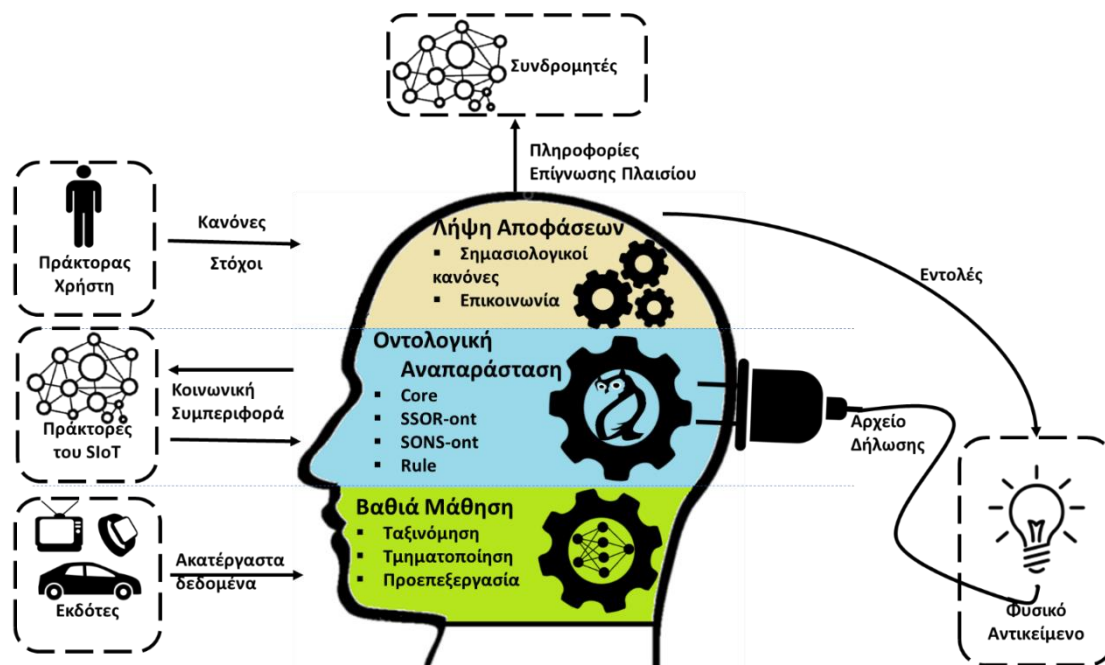
## 3.2 Γνωσιακοί Πράκτορες για το Κοινωνικό Διαδίκτυο των Πραγμάτων

Οι προτεινόμενοι γνωσιακοί πράκτορες λογισμικού για το Κοινωνικό Διαδίκτυο των Πραγμάτων με στόχο την επίγνωση πλαισίου, αποτελούνται από τρία στρώματα:

- Το στρώμα *Βαθιάς Μάθησης*, όπου πραγματοποιείται η αναγνώριση του πλαισίου.
- Το στρώμα *Οντολογικής Αναπαράστασης* των πληροφοριών και ελέγχου συμμόρφωσης των σημασιολογικών κανόνων.
- Το στρώμα *Λήψης Αποφάσεων*, όπου ενεργοποιούνται οι σημασιολογικοί κανόνες και λαμβάνει μέρος η επικοινωνία με τις υπόλοιπες οντότητες του Κοινωνικού Διαδικτύου των Πραγμάτων.

Όπως φαίνεται και στην Εικόνα 24, στο κάτω στρώμα εφαρμόζονται τεχνικές επεξεργασίας σήματος (προεπεξεργασία και τμηματοποίηση) και Βαθιάς Μάθησης με στόχο την εξαγωγή πληροφοριών πλαισίου από ακατέργαστα δεδομένα (π.χ., σήματα αισθητήρων). Στο μεσαίο στρώμα γίνεται η αναπαράσταση των πληροφοριών πλαισίου με χρήση οντολογιών, ενώ πραγματοποιείται και συλλογισμός επάνω στα οντολογικά δεδομένα. Επιπλέον, το συγκεκριμένο στρώμα μοντελοποιεί τις πληροφορίες αρχικοποίησης (δηλ., το *manifest file*, βλ. Ενότητα 3.5.1) της οντότητας που εκπροσωπεί ο πράκτορας λογισμικού (π.χ., μία έξυπνη λάμπα), τις κοινωνικές σχέσεις του πράκτορα και τους στόχους της οντότητας. Θα πρέπει να σημειωθεί πως το συγκεκριμένο στρώμα είναι υπεύθυνο και για τον έλεγχο

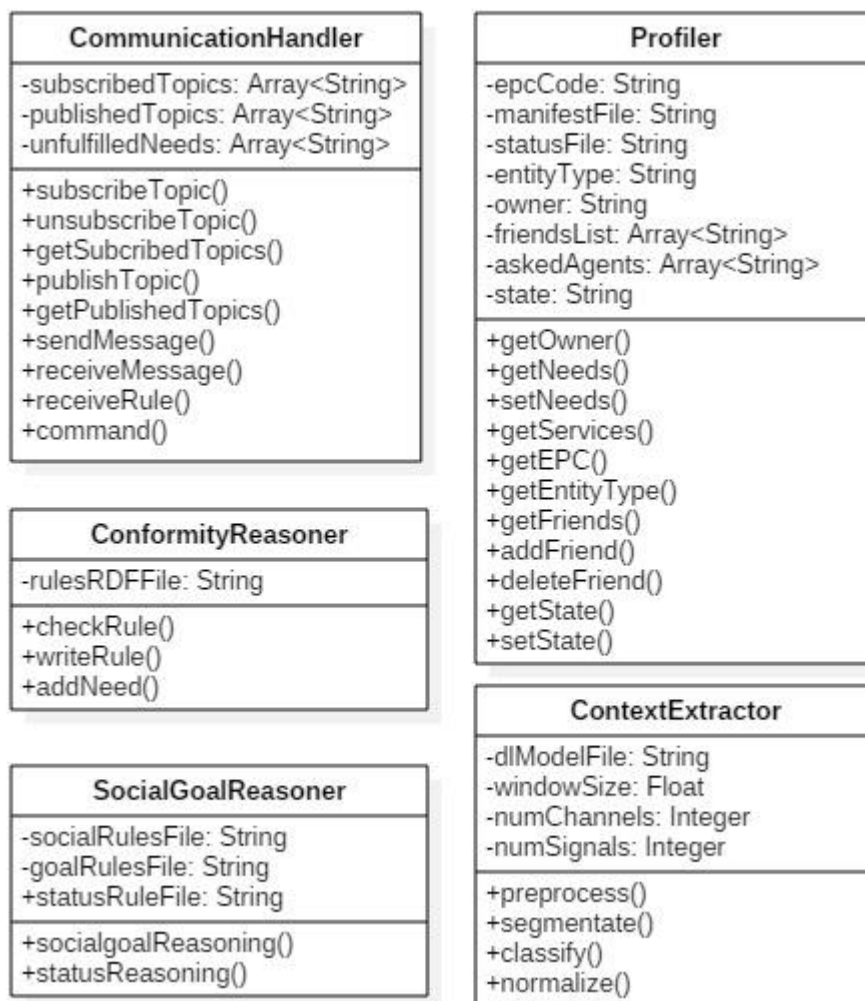
εγκυρότητας των κανόνων συμπεριφοράς που αφορούν (σ)την αντιπροσωπούμενη οντότητα. Τέλος, το τελευταίο στρώμα έχει ως αρμοδιότητες την επιβολή κανόνων, κάνοντας χρήση παραγωγικού συλλογισμού, και την επικοινωνία με τις υπόλοιπες φυσικές και μη οντότητες του SIoT οικοσυστήματος.



Εικόνα 24: Αρχιτεκτονική γνωσιακού πράκτορα επίγνωσης πλαισίου για το SIoT

### 3.3 Συστατικά στοιχεία λογισμικού του πράκτορα του SIoT

Το προτεινόμενο Κοινωνικό Δίκτυο αποτελείται από τους πράκτορες λογισμικού του Κοινωνικού Διαδικτύου των Πραγμάτων (SIoT Agents), οι οποίοι είναι σκεπτόμενες οντότητες, και έχει τρεις τύπους: *Device Agent*, *User Agent* και *Task Agent*. Κάθε τύπος SIoT Agent είναι μία πολυσύνθετη οντότητα και αποτελείται από πολλές κλάσεις/νήματα (threads), ώστε να επιτρέψει στην οντότητα που αντιπροσωπεύει να έχει κοινωνικές και γνωσιακές δραστηριότητες. Πιο συγκεκριμένα, τα στοιχεία λογισμικού του SIoT Agent είναι τα εξής (Εικόνα 25):



Εικόνα 25: Τα συστατικά στοιχεία λογισμικού του SIoT Agent

- ContextExtractor*: είναι υπεύθυνο για την επεξεργασία των ακατέργαστων δεδομένων. Τέτοιας μορφής είναι τα δεδομένα που παράγονται από αισθητήρες του IoT, όπως είναι δεδομένα από επιταχυνσιόμετρο ή γυροσκόπιο. Όπως φαίνεται και στην Εικόνα 24, το *ContextExtractor* βρίσκεται στο χαμηλότερο στρώμα αρχιτεκτονικής του SIoT Agent και βασίζεται σε έναν αλγόριθμο Βαθιάς Μάθησης για την εξαγωγή πληροφοριών πλαισίου. Συγκεκριμένα, πριν τροφοδοτηθούν τα δεδομένα στο μοντέλο, διέρχονται από τα στάδια προεπεξεργασίας και τμηματοποίησης, ενώ οι τιμές του κανονικοποιούνται. Έπειτα ο αλγόριθμος (Εικόνα 17) τα επεξεργάζεται και καταλήγει σε μία πιθανότητα ταξινόμησής τους σε μία ανθρώπινη δραστηριότητα. Τέλος, το *ContextExtractor* ενημερώνει τον *Profiler* για το αποτέλεσμα.
- Profiler*: πρόκειται για το κύριο συστατικό στοιχείο, και καθώς φέρει γνώση για όλα τα δεδομένα που αφορούν (σ)την οντότητα εκπροσώπησης, ανήκει στο στρώμα *Οντολογικής Αναπαράστασης* του

SIoT Agent. Επιπλέον, είναι υπεύθυνος για την ανάλυση του αρχείου δήλωσης (*manifest file*). Το εν λόγω αρχείο κάνει χρήση της γλώσσας *turtle* και περιέχει όλες τις απαραίτητες πληροφορίες, εκφρασμένες σε σημασιολογικές τριπλέτες. Πιο συγκεκριμένα, υποδηλώνει το όνομα, το μοναδικό αναγνωριστικό, τις ανάγκες, τις υπηρεσίες και τους Γονικούς Φίλους της έξυπνης οντότητας. Δημιουργείται κατά την κατασκευή του αντικειμένου και δεν μπορεί να αλλάξει μετά τη δημιουργία του (Ενότητα 3.5.1). Ο λόγος για τη δημιουργία αυτού του αρχείου είναι για να μεταβιβάσει όλες τις πληροφορίες στον SIoT Agent που θα αντιπροσωπεύει τη συγκεκριμένη οντότητα στο Κοινωνικό Δίκτυο των Πρακτόρων λογισμικού. Αυτές οι πληροφορίες δεν θα τροποποιηθούν ακόμη και αν το έξυπνο αντικείμενο αλλάξει ιδιοκτήτη (π.χ., πώληση σε άλλον χρήστη). Μετά την ανάγνωση του *manifest file*, ο *Profiler* παράγει ένα σχεδόν ίδιο αρχείο, το αρχείο κατάστασης (*status file*), το οποίο περιλαμβάνει τον ιδιοκτήτη της έξυπνης οντότητας, το είδος του πράκτορα λογισμικού (δηλ., αν πρόκειται για αντικείμενο ή εφαρμογή), τους φίλους του, καθώς και τα στοιχεία που περιλαμβάνει το *manifest file*. Ο *Profiler* ενημερώνει το αρχείο κατάστασης όταν έχει υπάρξει κάποια ανάγκη ή όταν έχει αλλάξει η κοινωνική κατάστασή του (π.χ., πρόσθεσε κάποιο φίλο, διέγραψε κάποιο φίλο, έκανε αίτημα φιλίας σε ένα άλλο SIoT Agent κ.λπ.).

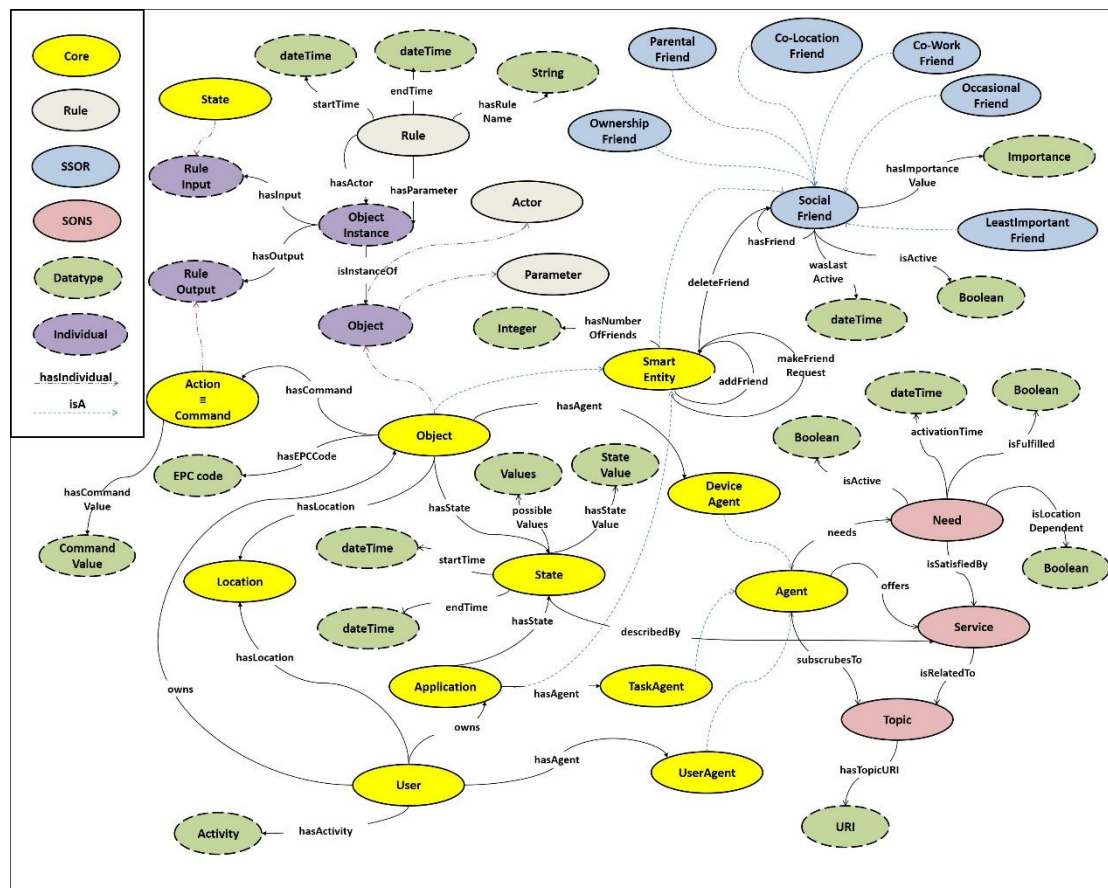
- *ConformityReasoner*: πρόκειται για το στοιχείο λογισμικού που εξασφαλίζει αν οι κανόνες που εισάγει ο χρήστης για τις συσκευές του δεν είναι αντικρουόμενοι/αντιφατικοί. Οι συγκεκριμένοι κανόνες συμπεριφοράς είναι εκφρασμένοι σε σημασιολογική μορφή κάνοντας χρήση της οντολογίας Rule (Ενότητα 3.4.4). Σε περίπτωση που ο κανόνας είναι αντιφατικός, τότε το *ConformityReasoner* ενημερώνει τον *CommunicationHandler*, ενώ στην αντίθετη περίπτωση γράφει τον κανόνα στο ηλεκτρονικό αποθετήριο κανόνων και εισάγει τις ανάγκες που προκύπτουν βάσει του καινούριου κανόνα στη λίστα *needsList*.
- *CommunicationHandler*: Εφόσον είναι μία κοινωνική οντότητα, ο SIoT Agent πρέπει να διαθέτει ένα συστατικό στοιχείο για να διαχειρίζεται τις συνομιλίες του, και ανήκει στο στρώμα λήψης αποφάσεων. Το *CommunicationHandler* (CH) είναι υπεύθυνο για την επικοινωνία με τους άλλους SIoT Agents (π.χ., User Agent) χρησιμοποιώντας το πρωτόκολλο μηνυμάτων FIPA (Foundation of Intelligent Physical Agents). Με αυτόν τον τρόπο το CH μπορεί να πραγματοποιεί ή να δέχεται αιτήματα φιλίας. Άλλο ένα καθήκον του CH είναι η αποστολή εντολών προς τη συσκευή που εκπροσωπεί, με στόχο την αλλαγή της κατάστασης της. Επιπλέον, στην περίπτωση δημιουργίας ενός νέου κανόνα από το χρήστη, ο

πράκτορας του στέλνει στο *SlOT Agent*, ο οποίος εκπροσωπεί τη συσκευή που θα επηρεάσει ο κανόνας, ένα μήνυμα που θα του παρέχει όλα τα στοιχεία του κανόνα σημασιολογικά ορισμένα. Το *CH* είναι επίσης υπεύθυνο για τη δημοσίευση θεμάτων και τη λήψη μηνυμάτων που αφορούν τον *SlOT Agent* (δηλ., από τα θέματα που είναι συνδρομητής).

- *Reasoner*: είναι το συστατικό στοιχείο συλλογισμού και λήψης αποφάσεων, ενεργώντας ως Διαχειριστής Σχέσεων (*Relationship Manager*) και ανήκει στο υψηλότερο επίπεδο της αρχιτεκτονικής. Όταν συμβαίνει ένα κοινωνικό συμβάν ή αλλάζει η κατάσταση ενός φίλου του πράκτορα λογισμικού, ενεργοποιείται το στοιχείο συλλογισμού. Πιο συγκεκριμένα, αναλύει τους αποθηκευμένους σημασιολογικούς κανόνες σχετικά με τη διαχείριση των σχέσεων του *SlOT Agent* και επιλέγει τις μελλοντικές του ενέργειες. Αυτές οι ενέργειες αφορούν (σ)την προσθήκη ενός φίλου, τη διαγραφή ενός φίλου, την αποστολή αιτήματος φιλίας ή την αποστολή φιλίας σε φίλο ενός φίλου (*Friend of a Friend; FoaF*). Αποφασίζει με βάση τις ανεκπλήρωτες ανάγκες της έξυπνης οντότητας, στην οποία αντιστοιχεί ο *SlOT Agent*, προκειμένου να τις εκπληρώσει και να μεγιστοποιήσει τη χρησιμότητά του. Επιπλέον, τα δεδομένα που πρέπει να λάβει υπόψη του το *Reasoner*, στις περισσότερες από τις προαναφερθείσες περιπτώσεις, είναι ο αριθμός των φίλων, και η παράμετρος της σπουδαιότητας (δηλαδή μία τιμή που απεικονίζει πόσο κερδίζει ένας *SlOT Agent* από μία σχέση του). Τέλος, το *Reasoner* τρέχει και τους κανόνες αυτοματισμού που αφορούν τη αντιπροσωπούμενη οντότητα. Για παράδειγμα, αν μία έξυπνη λάμπα πρέπει να ανάψει αν ανοίξει η πόρτα του δωματίου στο οποίο βρίσκεται, τότε κάθε φορά που συμβαίνει μία μεταβολή στην κατάσταση του ενεργοποιητή της πόρτας το *Reasoner* τρέχει το συγκεκριμένο κανόνα. Στην περίπτωση που αποφασίσει ότι πρέπει να προβεί σε κάποια αλλαγή στην κατάσταση του αντικειμένου, ενημερώνει το *CH* για να στείλει την εντολή.

### 3.4 Σημασιολογική Αναπαράσταση Σχέσεων

Όλες οι οντότητες του *SlOT* οικοσυστήματος έχουν τη δυνατότητα να αλληλεπιδρούν μεταξύ τους, εκμεταλλευόμενες το φαινόμενο του Μικρού Κόσμου. Η δυνατότητα αυτή οδηγεί σε ζητήματα διαλειτουργικότητας που πρέπει να επιλυθούν. Ένας κατάλληλος τρόπος αντιμετώπισης της ετερογενούς φύσης των έξυπνων συσκευών, αλλά και επίτευξης του γνωσιακού συλλογισμού στα δεδομένα/πληροφορίες είναι η χρήση των τεχνολογιών του Σημασιολογικού Ιστού.



Εικόνα 26: Η προτεινόμενη οντολογία για την επίτευξη γνωσιακών λειτουργιών στο Κοινωνικό Διαδίκτυο των Πραγμάτων.

Για τους παραπάνω λόγους, σχεδιάστηκαν τέσσερις οντολογίες:

- Core, κύρια οντολογία περιγραφής των έξυπνων οντοτήτων και των πρακτόρων λογισμικού.
- SSOR-Ont, οντολογία περιγραφής των κοινωνικών σχέσεων μεταξύ έξυπνων οντοτήτων.
- SONS-Ont, οντολογία περιγραφής των αναγκών και των υπηρεσιών των έξυπνων οντοτήτων.
- Rule, οντολογία περιγραφής των κανόνων αυτοματισμού για τις συσκευές του IoT.

Στην Εικόνα 26 παρουσιάζεται ένα συγκεντρωτικό σχήμα, το οποίο συνθέτουν οι παραπάνω οντολογίες. Συγκεκριμένα, οι κλάσεις της οντολογίας Core παρουσιάζονται με κίτρινο χρώμα, της SSOR-Ont με γαλάζιο, της SONS-Ont με ροζ, ενώ της οντολογίας Rule με γκριζο. Στις παρακάτω ενότητες παρουσιάζονται λεπτομερώς τα παραπάνω οντολογικά σχήματα, τα οποία σχεδιάστηκαν στο protέγ 4.3. [127].

### 3.4.1 Οντολογία Core

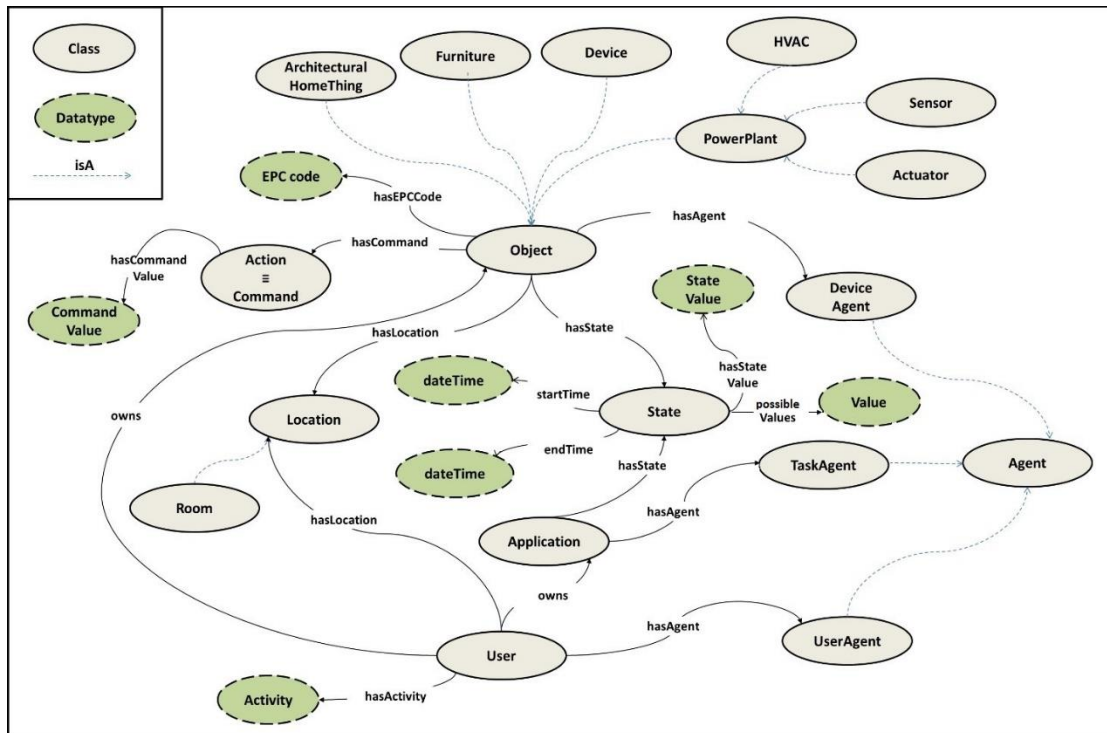
Σε αυτή την ενότητα παρουσιάζεται ένα οντολογικό σχήμα σχεδιασμένο κατά τρόπο που να μπορεί να υποστηρίξει πλήρως την περιγραφή του χώρου ενός έξυπνου σπιτιού, να εξυπηρετεί τη σημασιολογική επικοινωνία και τη διαλειτουργικότητα μεταξύ των συσκευών του *IoT*, και, τέλος, να ικανοποιήσει τις ανάγκες ενός πολυπρακτορικού συστήματος το οποίο θα παίρνει αποφάσεις βάσει παραγωγικού συλλογισμού (*MAS*). Η συγκεκριμένη οντολογία ονομάζεται οντολογία Κορμού (*Core ontology*) και πάνω σε αυτή «κουμπώνουν» οι υπόλοιπες οντολογίες, έτσι ώστε να επιτευχθεί η κοινωνικοποίηση και η αποτελεσματική διαχείριση των στόχων τους.

Η οντολογία *Core* είναι δομημένη για να ενισχύσει το όραμα του *IoT*, κυρίως όσον αφορά τα συστήματα οικιακού αυτοματισμού και βασίζεται σε προηγούμενες ερευνητικές δουλειές που αναφέρονται στην Ενότητα 2.2.2. Συγκεκριμένα, επωφελείται από το διαχωρισμό των οικιακών συσκευών σε καφέ και λευκές, καθώς και τις υποκατηγορίες των διαφόρων αισθητήρων και ενεργοποιητών που προτείνουν οι οντολογίες *DomoML-ent* [55] και *Dog-Ont* [53] αντιστοίχως. Επιπλέον, έχει σχεδιαστεί με τέτοιο τρόπο ώστε να επιτρέπει στους έξυπνους πράκτορες λογισμικού να χειρίζονται μέσω προδιαγεγραμμένων εντολών ή κανόνων τις λειτουργίες των συσκευών που τους αντιστοιχούν.

Οι κύριες οντότητες της οντολογίας *Core* παρουσιάζονται στην Εικόνα 27 και είναι οι εξής:

- *Object (Αντικείμενο)*: Σε αυτήν ανήκουν έξυπνα (συσκευές *IoT*) και μη αντικείμενα (π.χ., έπιπλα). Τα έξυπνα αντικείμενα απαρτίζονται από τις υποκλάσεις *PowerPlants*, *Device (BrownGood and WhiteGood)*, ενώ τα μη-έξυπνα από *ArchitecturalHomeThings* (π.χ., Πόρτα) and *Furniture*.
- *User (Χρήστης)*: Αντιπροσωπεύει το χρήστη/ιδιοκτήτη των έξυπνων αντικειμένων και εφαρμογών, ενώ οι οντότητες που του ανήκουν περιγράφονται από την Ιδιότητα Αντικειμένου *owns*.
- *Application (Εφαρμογή)*: Χαρακτηρίζει τις εφαρμογές που έχει στην κατοχή του ο Χρήστης.
- *Location (Τοποθεσία)*: Πρόκειται για τις περιοχές/δωμάτια ενός έξυπνου σπιτιού στις οποίες βρίσκονται (*hasLocation*) τα αντικείμενα. Η συγκεκριμένη κλάση επιτρέπει τη δημιουργία σχέσεων συνεγκατάστασης μεταξύ των έξυπνων οντοτήτων, καθώς και την περιγραφή τους με τέτοιο τρόπο ώστε να είναι βασισμένες στην τοποθεσία.

- *State (Κατάσταση)*: Η συγκεκριμένη κλάση αναπαριστά την κατάσταση στην οποία βρίσκεται ένα έξυπνο αντικείμενο. Για παράδειγμα, μία έξυπνη λάμπα μπορεί να έχει δύο καταστάσεις, *OnState* και *OffState*. Η τιμή μίας κατάστασης καθορίζεται από την Ιδιότητα Δεδομένων *hasStateValue*, ενώ οι πιθανές τιμές που μπορεί να πάρει από την Ιδιότητα Δεδομένων *possibleValues*.



Εικόνα 27: Παρουσίαση της οντολογίας Core.

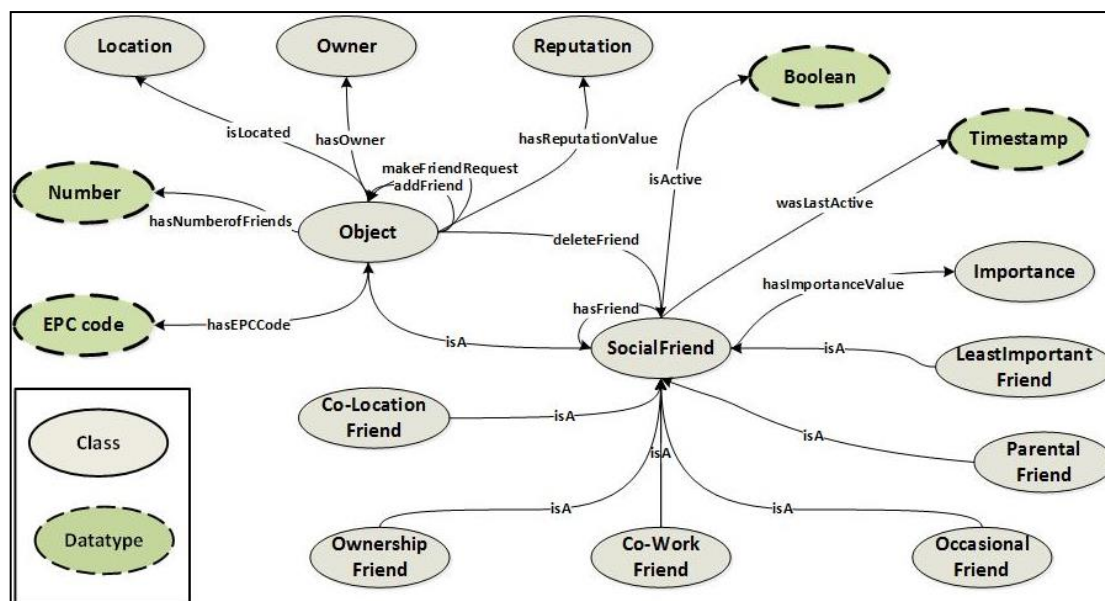
- *Εντολή (Command)*: Εδώ ανήκουν οι δράσεις/εντολές που απαιτούνται για να μεταβληθεί η κατάσταση ενός αντικειμένου. Η τιμή που μπορεί να πάρει μία εντολή χαρακτηρίζεται από την Ιδιότητα Δεδομένων *hasCommandValue*.
- *Agent (Πράκτορας)*: Οι έξυπνοι πράκτορες λογισμικού του προτεινόμενου ΣIoT οικοσυστήματος ανήκουν σε αυτήν την κλάση. Υποκλάσεις της είναι οι *DeviceAgent*, *UserAgent*, και *TaskAgent* που αντιπροσωπεύουν τις οντότητες *Object*, *User*, και *Application* αντιστοίχως, μέσω της Ιδιότητας Αντικειμένου *hasAgent*.
- *Activity (Δραστηριότητα)*: Πρόκειται για τις δραστηριότητες που εκτελούνται από τους ανθρώπους και αποτελεί μία πληροφορία πλαισίου. Για παράδειγμα, η υποκλάση *Running* χαρακτηρίζει το τρέξιμο.



### 3.4.2 Οντολογία SSOR-Ont

Η Οντολογία Περιγραφής των Σχέσεων του Κοινωνικού Έξυπνου Αντικειμένου (Social Smart Object's Relationships Ontology; SSOR-Ont) [44] αντιπροσωπεύει τους τύπους κοινωνικών σχέσεων μεταξύ των αντικειμένων όπως αυτές παρουσιάζονται στην [33]. Όπως φαίνεται στην Εικόνα 28, η κύρια κλάση της οντολογίας SSOR-Ont είναι η *SocialFriend* (Κοινωνικός Φίλος) που είναι ο χαρακτηρισμός μίας έξυπνης οντότητας. Περιέχει πέντε κύριες υποκατηγορίες:

- *OwnershipFriend* (Φίλος Ιδιοκτησίας): Απεικονίζει τη Σχέση Ιδιοκτησίας Αντικειμένου. Οι συσκευές που ανήκουν στον ίδιο ιδιοκτήτη θεωρούνται ότι έχουν αυτό το είδος της σχέσης, με στόχο την παροχή εξατομικευμένων υπηρεσιών.
- *Co-LocationFriend* (Φίλος Συνεγκατάστασης): Αντιπροσωπεύει τη Σχέση Συνεγκατάστασης Αντικειμένου. Αυτό το είδος της σχέσης μεταξύ συσκευών δημιουργείται μεταξύ αντικειμένων που τοποθετούνται στην ίδια περιοχή (έξυπνο σπίτι ή έξυπνη πόλη).
- *Co-WorkFriend* (Φίλος Συνεργασίας): Απεικονίζει τη Σχέση Συνεργασίας Αντικειμένου. Συσκευές που συνεργάζονται μεταξύ τους για έναν κοινό στόχο δημιουργούν αυτόν τον τύπο σχέσης.
- *OccasionalFriend* (Περιστασιακός Φίλος): Αυτή η σχέση μπορεί να δημιουργηθεί μεταξύ αντικειμένων που συναντιούνται περιστασιακά σε ένα δίκτυο Mobile Ad-Hoc Network (MANET), για παράδειγμα έξυπνα αυτοκίνητα σε ένα δίκτυο οχημάτων Ad Hoc (VANET) [128]. Αυτού του είδους οι σύνδεσμοι μπορεί να αποδειχθούν εξαιρετικά χρήσιμοι για την κάλυψη αναγκών που σχετίζονται όχι μόνο με το χώρο, αλλά και με το χρόνο. Για παράδειγμα, για να βρει ένα έξυπνο αυτοκίνητο τη βέλτιστη (ταχύτερη) διαδρομή μπορεί να ρωτήσει φίλους του, όπως άλλα αυτοκίνητα ή φανάρια.
- *ParentalFriend* (Γονεϊκός Φίλος): Αντιπροσωπεύει τη σχέση που ένα έξυπνο αντικείμενο έχει με άλλα έξυπνα αντικείμενα που ανήκουν στην ίδια παρτίδα παραγωγής. Οι γονεϊκές σχέσεις είναι χρήσιμες όταν πρόκειται για την αντιμετώπιση προβλημάτων λογισμικού που έχουν ήδη αντιμετωπίσει τα άλλα παρόμοια έξυπνα προϊόντα.

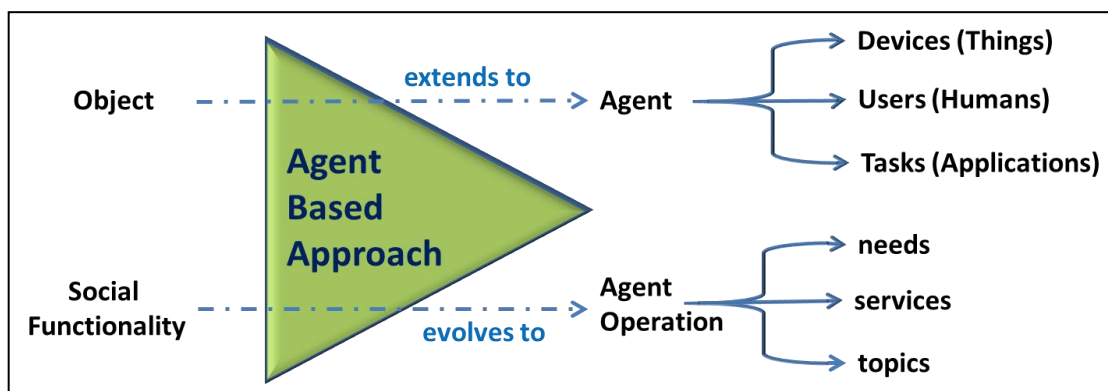


Εικόνα 28: Το σχήμα της οντολογίας SSOR-Ont [44].

Η κλάση *LeastImportantFriend* είναι η οντότητα που δηλώνει ποιος από τους φίλους του αντικειμένου είναι λιγότερο σημαντικός. Η Ιδιότητα Αντικειμένου (Object Property) *hasOwner* ορίζει τον ιδιοκτήτη του αντικειμένου, ενώ η *hasFriend* δηλώνει τη φιλία μεταξύ δύο στιγμιστύπων της κλάσης *SocialFriend*. Η ιδιότητα *makeFriendRequest* δηλώνει ότι ένα αντικείμενο θέλει να προσθέσει στον κατάλογο φίλων ένα άλλο κάνοντας αίτημα φιλίας. Η ιδιότητα *addFriend* προσθέτει ένα αντικείμενο στη λίστα φίλων, ενώ η *deleteFriend* διαγράφει έναν κόμβο από τη λίστα κοινωνικών σχέσεων. Οι Ιδιότητες Δεδομένων (Data Properties) που δημιουργήθηκαν αντιπροσωπεύουν πόσους φίλους έχει ένα αντικείμενο (*hasNumberOfFriends*), τον Ηλεκτρονικό Κωδικό Προϊόντος (Electronic Product Code; EPC) [129] που έχει (*hasEPCCode*), πότε ήταν ενεργή η σχέση μεταξύ δύο κόμβων (*wasActiveAt*), δηλαδή πότε αντάλλαξαν δεδομένα, και αν ένας κόμβος είναι ενεργός (*isActive*). Οι ενεργοί κόμβοι θεωρούνται ότι είναι κόμβοι που επικοινωνούν με το αντικείμενο για να παρέχουν την υπηρεσία σε αυτό. Επιπλέον, το *hasImportanceValue* ορίζει πόσο σημαντικός είναι ένας φίλος (*Importance*) και μπορεί να πάρει τρεις τιμές: *High* (Υψηλή), *Medium* (Μεσαία) και *Low* (Χαμηλή). Αυτές οι τιμές μπορούν να προκύψουν με χρήση τεχνικών μηχανικής μάθησης ή τους σημασιολογικούς κανόνες. Τέλος, η ιδιότητα *hasReputationValue* απεικονίζει τη φήμη που έχει ένα κοινωνικό αντικείμενο. Μπορεί να πάρει δύο τιμές: *Good* (Κακή) ή *Bad* (Καλή). Οι κόμβοι που έχουν κακή φήμη θεωρούνται κακόβουλοι ή μη συνεργατικοί.

Η οντολογία SSOR-Ont σχεδιάστηκε για να επισημάνει το είδος των σχέσεων που κάθε οντότητα έχει να διαχειριστεί. Στο πλαίσιο μίας προσέγγισης που λαμβάνει υπόψη την προστασία της ιδιωτικής ζωής, μόνο τα δύο έξυπνα αντικείμενα που έχουν δημιουργήσει μία σχέση γνωρίζουν αυτό το γεγονός, ενώ η σχέση παραμένει κρυφή από τα υπόλοιπα. Φυσικά, ο ιδιοκτήτης του αντικειμένου μπορεί να επιτρέψει, μέσω

της χρήσης πολιτικών, σε όλες τις συσκευές ή απλώς στους φίλους του να μάθουν τη λίστα φίλων του. Ωστόσο, στις περισσότερες περιπτώσεις του IoT, τα "Πράγματα", ακόμη και αν θεωρούνται έξυπνα (π.χ., έξυπνοι λαμπτήρες), δεν είναι σε θέση να παίρνουν πολύπλοκες αποφάσεις με βάση την επίτευξη των στόχων τους. Έτσι, η προτεινόμενη προσέγγιση βασίζεται σε ένα πολυπρακτορικό σύστημα, σε αντίθεση με την προσέγγιση [33]. Σύμφωνα με αυτή την προσέγγιση (Εικόνα 29), η κλάση *Object* (Αντικείμενο) επεκτείνεται στην κλάση *Agent* (Πράκτορας) και η κοινωνική λειτουργικότητα των αντικειμένων (*Social Functionality*) εξελίσσεται σε λειτουργίες του Πράκτορα που εξαρτώνται από τις ανάγκες (*needs*), τις υπηρεσίες (*services*) και τα θέματα (*topics*) που οι έξυπνοι οργανισμοί χρειάζονται, προσφέρουν και παράγουν ή εγγράφονται αντίστοιχα. Η παρούσα προσέγγιση που βασίζεται σε πράκτορες λογισμικού χρησιμοποιεί την υπολογιστική ισχύ που έχουν οι έξυπνοι πράκτορες (εξαιτίας της εγκατάστασής τους σε Υπολογιστικά Νέφη) για να πάρουν αποφάσεις σχετικά με τη διαχείριση φιλίας και με βάση αυτήν να επιτευχθεί η κάλυψη των στόχων τους (π.χ., ικανοποίηση μίας ανάγκης τους).

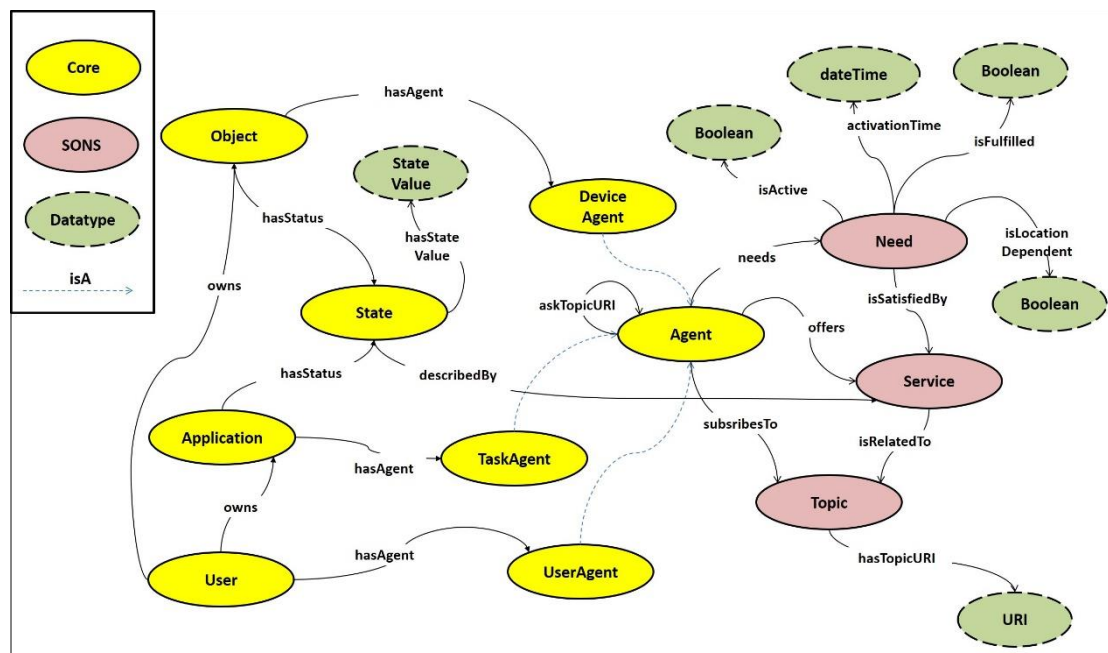


Εικόνα 29: Μετάβαση από το αρχικό SIoT στο SIoT πρότυπο βασισμένο σε έξυπνους πράκτορες λογισμικού [44].

### 3.4.3 Οντολογία SONS-Ont

Η Οντολογία Περιγραφής των Αναγκών και των Υπηρεσιών του Έξυπνου Αντικειμένου (Smart Object's Needs and Services Ontology; SONS-Ont) [44] περιγράφει σημασιολογικά τις ανάγκες ενός έξυπνου πράκτορα (δηλ., τις υπηρεσίες που καταναλώνει), και τις υπηρεσίες που παράγει (Εικόνα 30). Σε αντίθεση με την οντολογία SSOR-Ont, η κύρια οντότητα του SONS-Ont είναι ο *Agent* (Πράκτορας), δηλαδή κάνει χρήση της κλάσης *Agent* της οντολογίας Core. Ο τύπος του Πράκτορα μπορεί να είναι: *DeviceAgent*, *UserAgent* και *TaskAgent*, οι οποίοι αποτελούν τους πράκτορες λογισμικού των κλάσεων *Object* (Αντικείμενο), *User* (Χρήστης) και *Application* (Εφαρμογή) αντίστοιχα. Η σχέση μεταξύ των πρακτόρων λογισμικού και της οντότητας που αντιπροσωπεύουν απεικονίζεται από την Ιδιότητα Αντικειμένου *hasAgent*. Επιπλέον, το *Object* και ο *User* έχουν μία τοποθεσία που ορίζεται από την

Ιδιότητα Αντικειμένου *isLocated*. Ο Χρήστης επίσης μπορεί να κατέχει (*owns*) ή να δίνει εντολές (*commands*) σε ένα *Object* ή μία *Application*.



Εικόνα 30: Το σχήμα της οντολογίας SONS-Ont [44].

Η κλάση *Need* (Ανάγκη) καθορίζει τις ανάγκες που έχει ένας έξυπνος πράκτορας λογισμικού μέσω της Ιδιότητας Αντικειμένου *needs*, ενώ η κλάση *Service* (Υπηρεσία) καθορίζει τις υπηρεσίες που προσφέρει μέσω της Ιδιότητας Αντικειμένου *offers*. Οι ανάγκες ενός *Agent* μπορούν να έχουν τρεις Ιδιότητες Δεδομένων. Η ιδιότητα *isActive* περιγράφει εάν η ανάγκη είναι ενεργή (π.χ., ανάγκη στάθμευσης δεν υπάρχει εάν το αυτοκίνητο είναι ήδη σταθμευμένο), η ιδιότητα *activationTime* δηλώνει τη στιγμή ενεργοποίησης μίας ανάγκης (π.χ., η ανάγκη για χρονοπρογραμματισμό εξαρτάται από το χρόνο). Επιπλέον, η ιδιότητα *isFulfilled* δηλώνει εάν η ανάγκη έχει εκπληρωθεί από ένα *Topic* (Θέμα) και η ιδιότητα *isLocationDependent* δείχνει εάν μία ανάγκη σχετίζεται με μία θέση (π.χ., ανάγκη για περισσότερο φωτισμό σχετίζεται με το δωμάτιο που βρίσκεται ο χρήστης). Οι *Services* (Υπηρεσίες) ενός πράκτορα λογισμικού είναι τα θέματα που δημοσιεύει, σύμφωνα με το μοτίβο Δημοσίευσης-Συνδρομής. Αυτό περιγράφεται στην οντολογία SONS-Ont από την τριπλέτα *Service - isRelatedTo - Topic* και το Μοναδικό Αναγνωριστικό Πόρου (Uniform Resource Identifier; URI) του θέματος απεικονίζεται από την Ιδιότητα Δεδομένων *hasTopicURI*. Τέλος, σύμφωνα με τις ανάγκες του, ο πράκτορας λογισμικού εγγράφεται σε ένα θέμα. Αυτό επιτυγχάνεται μέσω σημασιολογικών κανόνων. Επιπλέον, η οντολογία SONS-Ont λαμβάνει υπόψη τις νέες ανάγκες που μπορούν να δημιουργηθούν με βάση την εξαγωγή των πληροφοριών πλαισίου, από τις τεχνικές Βαθιάς Μάθησης που είναι ενσωματωμένες στους έξυπνους πράκτορες, καθώς και από το συλλογισμό επάνω στους κανόνες αυτοματισμού με βάση την οντολογία Rule. Συνεπώς, νέες

σχέσεις συνεργασίας μπορούν να διαμορφωθούν δυναμικά έχοντας ως γνώμονα τις ανθρώπινες ανάγκες.

### 3.4.4 Οντολογία Rule

Για τις πλατφόρμες IoT που προσφέρουν στους χρήστες τους τη δυνατότητα να δημιουργήσουν τους δικούς τους κανόνες, ο εντοπισμός των αντικρουόμενων κανόνων αποτελεί ένα κρίσιμο ζήτημα. Οι κανόνες που ελέγχουν τις συσκευές εξαναγκάζονται από τους πράκτορες της συσκευής να έχουν διαφορετικούς στόχους, και αυτό οδηγεί στην εμφάνιση αντιφατικών καταστάσεων. Στον κόσμο του IoT, όπου ο ιδιοκτήτης ενός έξυπνου αντικειμένου θέλει να αυτοματοποιήσει τη συμπεριφορά των συσκευών του σε τομείς όπως η ασφάλεια, η ποιότητα ζωής και η εξοικονόμηση ενέργειας, ενδέχεται να συμβούν πολλές συγκρούσεις. Έτσι, παρόλο που οι κανόνες του μπορεί να είναι καλά γραμμένοι, υπάρχει περίπτωση να είναι αμοιβαία αντικρουόμενοι. Για παράδειγμα, ο χρήστης μπορεί να δημιουργήσει έναν κανόνα εξοικονόμησης ενέργειας που αναγκάζει τον πράκτορα λογισμικού μίας έξυπνης τηλεόρασης να την απενεργοποιήσει αν μία λάμπα είναι ενεργοποιημένη στο δωμάτιο, αλλά με βάση έναν άλλο κανόνα ο ίδιος πράκτορας λογισμικού πρέπει να ενεργοποιήσει την τηλεόραση σε μία συγκεκριμένη χρονική στιγμή, καθώς ο χρήστης θέλει να παρακολουθήσει την αγαπημένη του εκπομπή. Εάν υπάρχει ανοικτή λάμπα την ώρα της εκπομπής αυτοί οι δύο κανόνες συγκρούονται.

Το συστατικό στοιχείο λογισμικού *ConformityReasoner* του SIoT Agent είναι υπεύθυνο για την ανίχνευση πιθανών συγκρούσεων μεταξύ των κανόνων. Όταν δημιουργείται ένας νέος κανόνας, το *CommunicationHandler* λαμβάνει ένα μήνυμα, από τον User Agent, που τον ενημερώνει για αυτή την άφιξη αυτού του κανόνα, και αυτό με τη σειρά του το προωθεί στο *ConformityReasoner*. Σκοπός του είναι να ελέγξει εάν υπάρχει πιθανή σύγκρουση μεταξύ αυτού του κανόνα και των υπάρχοντων κανόνων που αφορούν την έξυπνη οντότητα με την οποία συνδέεται ο πράκτορας λογισμικού. Ο προτεινόμενος μηχανισμός εκτελείται offline.

Ενώ στην [37] είχε προταθεί ένας κεντροποιημένος μηχανισμός ανίχνευσης συγκρούσεων, η παρούσα διδακτορική διατριβή ακολουθεί μία αποκεντρωμένη προσέγγιση. Καθώς ο αριθμός  $n$  των έξυπνων αντικειμένων αυξάνεται και κάθε ένας από αυτούς συμπεριφέρεται με βάση έναν αριθμό κανόνων  $m$ , ο χώρος αναζήτησης αυξάνεται εκθετικά,  $m^n$ . Ως εκ τούτου, οι τεχνικές «κλαδέματος» θα πρέπει να υιοθετηθούν. Όπως αναφέρεται στο Ενότητα 3.3, η ανίχνευση των συγκρούσεων είναι η κύρια υπηρεσία του στοιχείου λογισμικού *ConformityReasoner*, το οποίο ελέγχει μόνο τους κανόνες που αφορούν τον πράκτορα της συσκευής του. Συνεπώς το ζήτημα της εύρεσης μίας σύγκρουσης σπάει σε πολλά υποπροβλήματα. Μία άλλη τεχνική «κλαδέματος» που ακολουθείται είναι αυτή του ελέγχου των συγκρούσεων κανόνα μόνο σε περιπτώσεις όπου υπάρχει ένα χρονικό διάστημα στο οποίο αυτοί οι





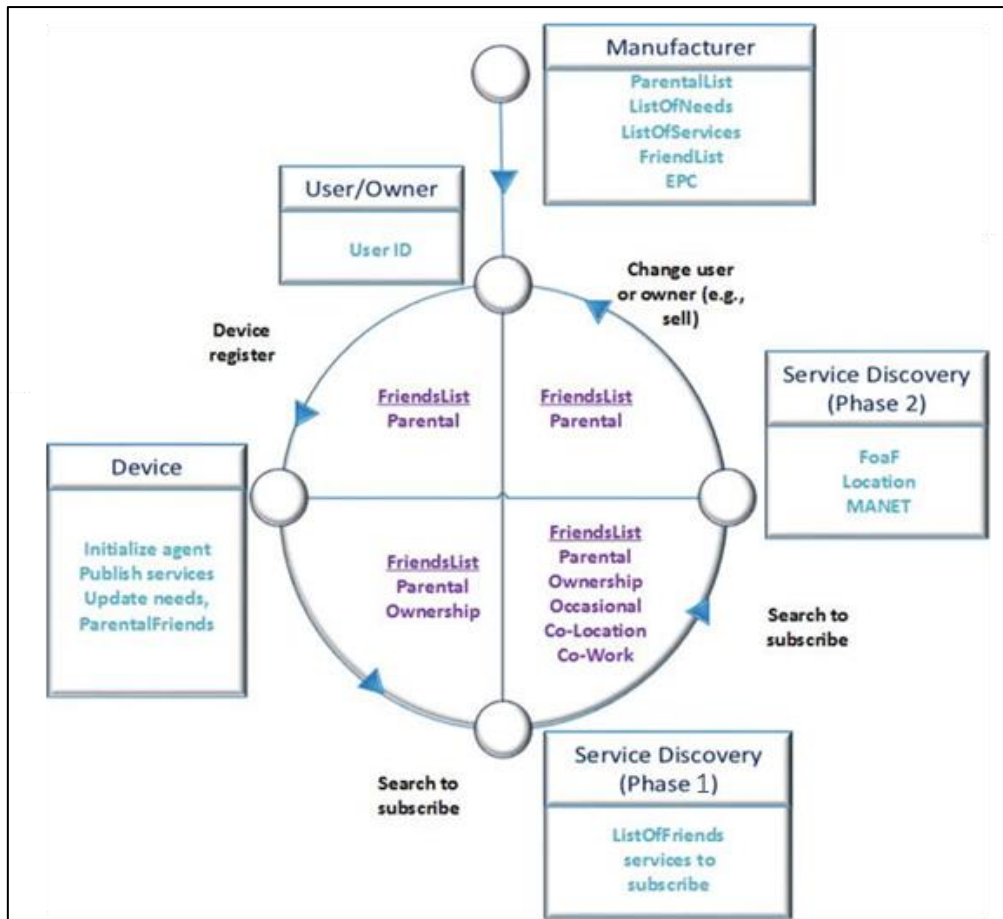
ισχύουν ώστε να γίνει ενεργός. Οι παράμετροι/συνθήκες του κάθε κανόνα είναι και αυτές στιγμιότυπα κάποιων αντικειμένων με βάση κάποιον κανόνα και έχουν τις δικές τους τιμές κατάστασης. Τέλος, το γεγονός ότι μία τοποθεσία μπορεί να έχει οριστεί για ένα έξυπνο αντικείμενο εκφράζεται μέσω της ιδιότητας *hasLocation*.

### 3.5 Διαδικασία Κοινωνικοποίησης Αντικειμένου

Η κοινωνικοποίηση ενός έξυπνου αντικειμένου είναι μία προκαθορισμένη διαδικασία, βασισμένη στο γνωσιακό πράκτορα λογισμικού που κάνει χρήση των τεχνολογιών του Σημασιολογικού Ιστού, και του ασύγχρονου προτύπου ανταλλαγής μηνυμάτων Δημοσίευσης-Συνδρομής [45]. Οι παραπάνω τεχνολογίες όχι μόνο ενισχύουν την ικανότητα του έξυπνου αντικειμένου να εξυπηρετεί τις ανάγκες του ιδιοκτήτη του, επιτυγχάνοντας τους στόχους του, αλλά και βοηθούν στην επικοινωνία με άλλες έξυπνες οντότητες του ΙοΕ. Στις επόμενες ενότητες περιγράφεται λεπτομερώς ο κύκλος ζωής του έξυπνου αντικειμένου καθώς και το πώς εφαρμόζεται το μοτίβο Pub-Sub στην προτεινόμενη προσέγγιση.

#### 3.5.1 Κύκλος Ζωής του Έξυπνου Αντικειμένου

Ο κύκλος ζωής μίας συσκευής ξεκινά όταν ένας κατασκευαστής τη δημιουργήσει και της αποδώσει ένα μοναδικό κωδικό, που ονομάζεται EPC. Αυτός ο κωδικός θα χρησιμοποιηθεί για τον εντοπισμό της συσκευής, και δεν θα δοθεί σε καμία άλλη συσκευή στον κόσμο. Εκτός από τον EPC, ο κατασκευαστής θα δημιουργήσει ένα *αρχείο δήλωσης (manifest file)* για κάθε συσκευή που περιέχει συγκεκριμένες πληροφορίες για τη συσκευή. Συγκεκριμένα, αυτές οι πληροφορίες περιλαμβάνουν μία λίστα με τις συσκευές που μοιράζονται την ίδια παρτίδα παραγωγής από τον ίδιο κατασκευαστή. Συνεπώς, οι συγκεκριμένες συσκευές εμφανίζουν μεταξύ τους μία Γονική Σχέση Αντικειμένου (POR), και ανήκουν στη λίστα *ParentalFriends* της συσκευής. Επιπλέον, θα δημιουργηθεί μία λίστα που περιγράφει τις ανάγκες της συσκευής, που ονομάζεται *ListOfNeeds* μαζί με μία λίστα που περιγράφει τους πόρους/υπηρεσίες που μπορεί να προσφέρει αυτή η συσκευή, η οποία ονομάζεται *ListOfServices*. Θα δημιουργηθεί, επίσης, η *FriendList*, ένας κατάλογος που περιέχει τους φίλους της συσκευής, αλλά σε αυτό το σημείο του κύκλου ζωής (Εικόνα 32) θα συμπληρωθεί με τις ίδιες πληροφορίες που βρίσκονται στην *ParentalFriends*, καθώς το αντικείμενο δεν έχει ακόμα αναπτύξει άλλου είδους φιλίες. Το *αρχείο δήλωσης* είναι αμετάβλητο και δεν μπορεί να αλλάξει από τον πράκτορα λογισμικού ή το χρήστη.



Εικόνα 32: Κύκλος ζωής του έξυπνου αντικειμένου [44].

```
@prefix sons: <http://consert.teipir.gr/research/results/consert-in-c-computing/ontologies/sons-ont.owl#> .
@prefix ssor: <http://consert.teipir.gr/research/results/consert-in-c-computing/ontologies/ssor-ont.owl#> .

ssor:activityTracker a ssor:Object ;
  sons:offers sons:caloriesBurned, sons:heartRateMeasuring, sons:stepMeasuring ;
  ssor:hasEPCCode "01.1234567.891011.001122DBA" ;
  sons:hasFriend ssor:object1, ssor:object2 .

sons:heartRateMeasuring a sons:Service .

sons:stepMeasuring a sons:Service .

sons:caloriesBurned a sons:Service .

ssor:object1
  a ssor:ParentalFriend ;
  ssor:hasEPCCode "01.1234567.891011.001122DBB" .

ssor:object2
  a ssor:ParentalFriend ;
  ssor:hasEPCCode "01.1234567.891011.001122DBC" .
```

Εικόνα 33: Το αρχείο δήλωσης μίας φορητής συσκευής (activity tracker) σε μορφή turtle.

Η Εικόνα 33 παρουσιάζει το αρχείο δήλωσης μίας φορητής συσκευής τύπου activityTracker (μετρητής δραστηριότητας). Ο activityTracker έχει ως EPC τον κωδικό "01.1234567.891011.001122DBA" και προσφέρει τρεις τύπους υπηρεσιών, συγκεκριμένα: α) StepMeasuring, β) HeartRate, και γ) CaloriesOut. Επιπλέον, το



αρχείο δήλωσης περιέχει και δύο γονεϊκούς φίλους: α) object1, και β) object2, μαζί με τους κωδικούς EPC τους.

Στη συνέχεια, η συσκευή πρόκειται να πωληθεί σε έναν χρήστη που θα είναι υπεύθυνος για αυτήν, και ο αντιστοίχος *UserAgent* θα την εγγράψει στο αρχείο του. Ο συγκεκριμένος χρήστης θα καταχωρηθεί ως ιδιοκτήτης της συσκευής. Οι χρήστες θα έχουν και αυτοί μοναδικά αναγνωριστικά χρήστη, τα οποία θα είναι μοναδικοί κωδικοί που θα αντιστοιχούν αποκλειστικά σε αυτούς. Επιπλέον, οι *UserAgents* θα έχουν τις δικές τους λίστες αναγκών, υπηρεσιών και φίλων (*ListOfNeeds*, *ListOfServices* και *FriendList*) που θα συμπληρώνονται με βάση τις πληροφορίες του χρήστη. Θα πρέπει να σημειωθεί πως ο τρέχων χρήστης της συσκευής ενδέχεται να μην είναι ο κάτοχος, καθώς η συσκευή μπορεί να μοιραστεί και με άλλους χρήστες (π.χ., μία ηλεκτρονική ζυγαριά).

Μετά την καταχώρηση της συσκευής στον τομέα του χρήστη (ή του κατόχου), ακολουθεί η δημιουργία του *DeviceAgent* (Εικόνα 32). Αυτός ο πράκτορας θα εξαγάγει τους πόρους/υπηρεσίες από το αρχείο δήλωσης του έξυπνου αντικειμένου και θα τους/τις δημοσιεύσει αργότερα, ως θέματα, στο στοιχείο λογισμικού Broker του προτύπου Pub-Sub, το οποίο θα είναι υπεύθυνο για την ανανέωση των πληροφοριών σχετικά με την ηλεκτρονική διεύθυνση των πόρων/υπηρεσιών και τη διαθεσιμότητά τους. Είναι, επίσης, καθήκον του *DeviceAgent* να ενημερωθεί σχετικά με τις λίστες *ListOfNeeds* και *ParentalFriends* της συσκευής μέσω ενός ερωτήματος SPARQL (Εικόνα 34). Έχοντας στην κατοχή του αυτές τις πληροφορίες, ο Πράκτορας Συσκευής, δημιουργεί το αρχείο κατάστασης.

```
PREFIX ssor: <http://consert.teipir.gr/research/results/consert-in-c-computing/ontologies/ssor-ont#>
PREFIX sons: <http://consert.teipir.gr/research/results/consert-in-c-computing/ontologies/sons-ont#>

SELECT ?need ?service ?friend ?epccode
WHERE {
    ?object a ssor:Object ;
           sons:offers ?service ;
           sons:needs ?need ;
           ssor:hasFriend ?friend .
    ?friend ssor:hasEPCCode ?epccode . }
```

Εικόνα 34: Ερώτημα SPARQL που επιστρέφει όλες τις υπηρεσίες, τις ανάγκες, και τους γονεϊκούς φίλους μαζί με τους EPC κωδικούς τους, από το αρχείο δήλωσης μίας συσκευής [44].

Ύστερα, ξεκινάει η πρώτη φάση της διαδικασίας ανακάλυψης υπηρεσιών/κάλυψης αναγκών, όπου ο Πράκτορας Συσκευής επικοινωνεί με τον αντίστοιχο Πράκτορα Χρήστη για να του αποστείλει ο δεύτερος τα αντικείμενα/εφαρμογές που κατέχει, και ο Πράκτορας Συσκευής τους στέλνει ένα αίτημα φιλίας. Στη συνέχεια, ο πράκτορας λογισμικού θα χρησιμοποιήσει τις νέες σχέσεις φιλίας (*OwnershipFriends*) για να διερευνήσει εάν κάποιος φίλος του προσφέρει στη *ListOfServices* τους συγκεκριμένους πόρους που απαιτούνται από τη συσκευή που αντιπροσωπεύει. Αν η αναζήτηση είναι επιτυχής και εφόσον του

επιτραπεί η πρόσβαση, θα εγγραφεί στο συγκεκριμένο θέμα για να αρχίσει να χρησιμοποιεί τους προσφερόμενους πόρους. Για παράδειγμα, αν ένας χρήστης χρειάζεται φωτισμό στο δωμάτιο, στο οποίο βρίσκεται, για να μπορέσει να διαβάσει, εισάγει μέσω μίας κατάλληλης διεπαφής στη *ListOfNeeds* το φωτισμό και ο *UserAgent* ψάχνει για μία συσκευή που παρέχει φως (και περιλαμβάνεται στην *ListOfServices* της συσκευής) στην ίδια τοποθεσία.

Δεδομένου ότι η ανακάλυψη των υπηρεσιών διαδραματίζει καίριο ρόλο στον κύκλο ζωής ενός έξυπνου αντικειμένου, θα πραγματοποιηθεί και μία δεύτερη φάση, στην περίπτωση που η πρώτη δεν είναι απόλυτα επιτυχής. Αυτή η φάση ενεργοποιείται όταν το αντικείμενο, δεν βρήκε έναν απαραίτητο πόρο (δηλαδή, υπάρχει η τριπλέτα *need - isFulfilled - false* στο *status file*), παρότι ήρθε σε επαφή με τους φίλους του χρήστη, και συνεπώς πρέπει να συνεχίσει να ψάχνει για τη συσκευή/εφαρμογή που προσφέρει το συγκεκριμένο θέμα. Σε αυτήν την προσπάθεια, το επόμενο βήμα είναι να δημιουργήσουν οι πράκτορες λογισμικού σχέσεις φιλίας με τους κόμβους που βρίσκονται στην *FriendList* των φίλων τους (δηλαδή, τους φίλους των φίλων τους), επεκτείνοντας την αναζήτηση βαθύτερα στο *SIoT* περιβάλλον. Άλλες πιθανές λύσεις για την εύρεση υπηρεσίας σε αυτή τη φάση περιλαμβάνουν την αποστολή αιτημάτων φιλίας σε συσκευές που βρίσκονται γεωγραφικά κοντά τους (*Co-LocationFriends*) ή αξιοποιούν παραδοσιακούς μηχανισμούς που χρησιμοποιούνται σε περιβάλλοντα Ομότιμων Κόμβων (*Peer-to-Peer*; *P2P*) και στα Αυτοοργανούμενα Δίκτυα (*Ad-Hoc Networks*) (π.χ., μέθοδοι πλημμύρας ή τυχαίοι περιπατητές).

Υπάρχει, επίσης, η περίπτωση όπου μία συσκευή δημιουργεί περιστασιακές σχέσεις (*OccasionalFriends*) με άλλες συσκευές/εφαρμογές που δεν ανήκουν στον ίδιο χρήστη, αλλά τις συναντάει συχνά σε κάποιο σημείο του κύκλου ζωής του. Όπως για παράδειγμα, συμβαίνει στην περίπτωση που ένα αυτοκίνητο, το οποίο κινείται σε μία μακρινή περιοχή, συναντά ένα άλλο όχημα που κινείται προς την αντίθετη κατεύθυνση. Τα δύο αυτοκίνητα μπορούν να δημιουργήσουν μία προσωρινή σχέση για να ανταλλάξουν πληροφορίες σχετικά με τη διαδρομή και να ενημερωθούν για τις καιρικές συνθήκες. Αυτές οι σχέσεις αντικειμένου στις περισσότερες περιπτώσεις είναι προσωρινές και δεν διαρκούν πολύ.

Στην Εικόνα 35, παρουσιάζεται ένα παράδειγμα *αρχείου κατάστασης* όπου καταγράφονται στοιχεία μίας έξυπνης λάμπας (*lamp*) σε μορφή *turtle*. Πιο συγκεκριμένα, η έξυπνη λάμπα ανήκει (*hasOwner*) στο χρήστη *John* και έχει δύο φίλους: α) τον *John*, και β) έναν αισθητήρα παρουσίας (*presenceSensor*). Ο αισθητήρας παρουσίας, προσφέρει την υπηρεσία *presenceDetectionService* στην οποία έχει εγγραφεί η λάμπα, καθώς την είχε ανάγκη. Συνεπώς, όπως φαίνεται και στο αρχείο *sons:presenceDetectionNeed - sons:isFulfilled - true*, η ανάγκη του για εντοπισμό παρουσίας είναι εκπληρωμένη. Θα πρέπει να σημειωθεί ότι, στο

συγκεκριμένο αρχείο κατάστασης, έχουν εγγραφεί και κάποια επιπλέον στοιχεία (εκτός από τις ανάγκες και τις υπηρεσίες που εμπεριέχονται και στο αρχείο δήλωσης), όπως ο βαθμός σημαντικότητας των φίλων της λάμπας (*ssor:presenceSensor - ssor:hasImportanceValue - 0.8388977*), ο συνολικός αριθμός των σχέσεων της (*ssor:lamp - ssor:hasNumberOfFriends - 2*), τα αιτήματα φιλίας που έχει δεχθεί (*ssor:presenceSensor - ssor:makeFriendRequest - ssor:lamp*) και ο χρόνος που δέχθηκε τα αιτήματα (*ssor:presenceSensor - ssor:makeFriendRequestAt - 1504521760787*).

Τέλος, κάθε φορά που ο χρήστης/ιδιοκτήτης πουλά τη συσκευή, ο κύκλος ολοκληρώνεται και πραγματοποιείται μία αλλαγή στον ιδιοκτήτη. Πρέπει να σημειωθεί ότι ο πράκτορας λογισμικού που αντιπροσώπευε το αντικείμενο παύει να υφίσταται και όλες οι κοινωνικές σχέσεις που είχε δημιουργήσει διαγράφονται (εκτός από τις γονικές φιλίες, καθώς περιλαμβάνονται στο *manifest file*).

```
@prefix sons: <http://consert.teipir.gr/research/results/consert-in-c-computing/ontologies/sons-ont.owl#> .
@prefix ssor: <http://consert.teipir.gr/research/results/consert-in-c-computing/ontologies/ssor-ont.owl#> .

ssor:lamp a ssor:Object ;
  sons:offers sons:luminocityService ;
  sons:produces sons:luminocityTopic ;
  sons:needs sons:presenceDetectionNeed ;
  ssor:hasEPCCode "01.1234567.891011.001133DBA" ;
  ssor:hasFriend ssor:presenceSensor, ssor:John ;
  ssor:hasNumberOfFriends 2.000000e+0 ;
  sons:subscribesTo sons:presenceDetectionTopic ;
  ssor:hasOwner ssor:John .

sons:luminocityTopic a sons:Topic ;
  sons:hasTopicURI "http://localhost:8080/user/123456789/object/01.1234567.891011.001122DBB/topic/luminocity" .

sons:luminocityService a sons:Service ;
  sons:isRelatedTo sons:luminocityTopic .

ssor:presenceSensor a ssor:OwnershipFriend, ssor:LeastImportantFriend ;
  ssor:hasImportanceValue 8.389077e-1 ;
  sons:offers sons:presenceDetectionService ;
  ssor:makeFriendRequest ssor:lamp ;
  ssor:makeFriendRequestAt 1504521760787 .

sons:presenceDetectionNeed a sons:Need ;
  sons:isFulfilled "true" ;
  sons:isSatisfiedBy sons:presenceDetectionService .

sons:presenceDetectionService a sons:Service ;
  sons:isRelatedTo sons:presenceDetectionTopic .

sons:presenceDetectionTopic a sons:Topic ;
  sons:hasTopicURI "http://localhost:8080/user/123456789/object/01.1234567.891011.004422DBB/topic/presenceDetection" .
```

Εικόνα 35: Το αρχείο κατάστασης μιας έξυπνης λάμπας (lamp) σε μορφή turtle.

### 3.5.2 Μηχανισμός Δημοσίευσης-Συνδρομής για το Κοινωνικό Διαδίκτυο των Πραγμάτων

Το μοτίβο ανταλλαγής μηνυμάτων που επιλέχθηκε για την παραγωγή και την κατανάλωση των πόρων του SIoT είναι το Pub-Sub. Το κύριο πλεονέκτημά του είναι ότι μπορεί να ωθήσει και να τραβήξει δεδομένα μεταξύ πολλών ετερογενών υπολογιστικών κόμβων ασύγχρονα μέσω της χρήσης ενός Μεσίτη (Broker). Πιο συγκεκριμένα, ο Μεσίτης επιτρέπει τη δημιουργία θεμάτων και την κατανάλωσή τους από τους ενδιαφερόμενους SIoT Agents. Δεδομένου ότι στο προτεινόμενο SIoT, η

επικοινωνία μεταξύ των οντοτήτων του πραγματοποιείται αυτόνομα, χωρίς την ανάγκη ανθρώπινης παρεμβολής, οι συσκευές, οι εφαρμογές και οι χρήστες χρησιμοποιούν το πρότυπο μηνυμάτων Pub-Sub, όπου οι Εκδότες (Publishers) ενημερώνουν το Μεσίτη για τις υπηρεσίες που προσφέρουν, καταλήγοντας στη δημιουργία αντίστοιχων θεμάτων, ενώ οι Συνδρομητές (Subscribers) λαμβάνουν πληροφορίες/δεδομένα από το Μεσίτη για τα Θέματα (Topics) που τους αφορούν.

Όπως αναφέρθηκε παραπάνω, η *ListOfServices* εκτίθεται από το Μεσίτη ως Θέματα. Στην παρούσα διατριβή, ακολουθείται μία προσέγγιση βασισμένη σε πόρους, όπως και στα περισσότερα RESTful APIs, για τη διάδοση των θεμάτων. Στην προτεινόμενη ιεραρχία, πρώτα εμφανίζεται η ηλεκτρονική διεύθυνση του Μεσίτη, μετά το αναγνωριστικό του χρήστη που κατέχει το αντικείμενο, ύστερα ο κωδικός EPC του αντικειμένου και τέλος το όνομα του Θέματος. Συγκεκριμένα, η δομή του URI ενός Θέματος έχει ως εξής: *{broker's electronic address}/user/{user\_ID}/object/{epc\_code}/topic/{topic\_name}*. Για παράδειγμα, εάν ένας χρήστης διαθέτει έναν αισθητήρα θερμοκρασίας και ο συγκεκριμένος χρήστης έχει ως μοναδικό αναγνωριστικό το 123456789, οι τιμές της θερμοκρασίας του περιβάλλοντος του αισθητήρα θα δημοσιευθούν ως: *http://localhost:8080/user/123456789/object/01.1234567.891011.001122DBB/topic/temperature*.

Ακολουθεί ένα παράδειγμα που αποσαφηνίζει τη λειτουργία του προτύπου ανταλλαγής μηνυμάτων Pub-Sub σε ένα περιβάλλον SIoT. Έστω, λοιπόν, ότι ένας άνθρωπος αγοράζει ένα έξυπνο ρολόι. Αφού καταγράψει τη συσκευή στον τομέα του και δημιουργηθεί ο πράκτορας της συσκευής για το ρολόι, ο πράκτορας του χρήστη ενημερώνει τον πράκτορα της εν λόγω συσκευής ότι ο χρήστης θέλει να γνωρίζει τις καιρικές συνθήκες και την κίνηση στους δρόμους. Ο πράκτορας της συσκευής, πρώτα δημοσιεύει στο Μεσίτη τις υπηρεσίες που μπορεί να προσφέρει το ρολόι (π.χ., την ώρα) και στη συνέχεια αναζητά φίλους που θα προσφέρουν μέσω Θεμάτων τις απαραίτητες υπηρεσίες για την κάλυψη των αναγκών του. Για παράδειγμα, αν κάνει φίλο τον τοπικό μετεωρολογικό σταθμό, θα μπορεί να λαμβάνει δεδομένα σχετικά με τις τρέχουσες καιρικές συνθήκες, αφού θα είναι Συνδρομητής στη συγκεκριμένη υπηρεσία.

## 4 Αναπτυχθέντα εργαλεία λογισμικού για το Κοινωνικό Διαδίκτυο των Πραγμάτων

Για την υλοποίηση και την αξιολόγηση της προτεινόμενης λύσης για το Κοινωνικό Διαδίκτυο των Πραγμάτων, αναπτύχθηκαν δύο εργαλεία λογισμικού. Το πρώτο ονομάζεται Sem-SHUI και αποτελεί τη διεπαφή του χρήστη με το οικοσύστημα του SIoT, δηλαδή τον Πράκτορα Χρήστη, ενώ το δεύτερο είναι ένας προσομοιωτής του προτεινόμενου SIoT περιβάλλοντος και ονομάζεται ASSIST.

### 4.1 Sem-SHUI

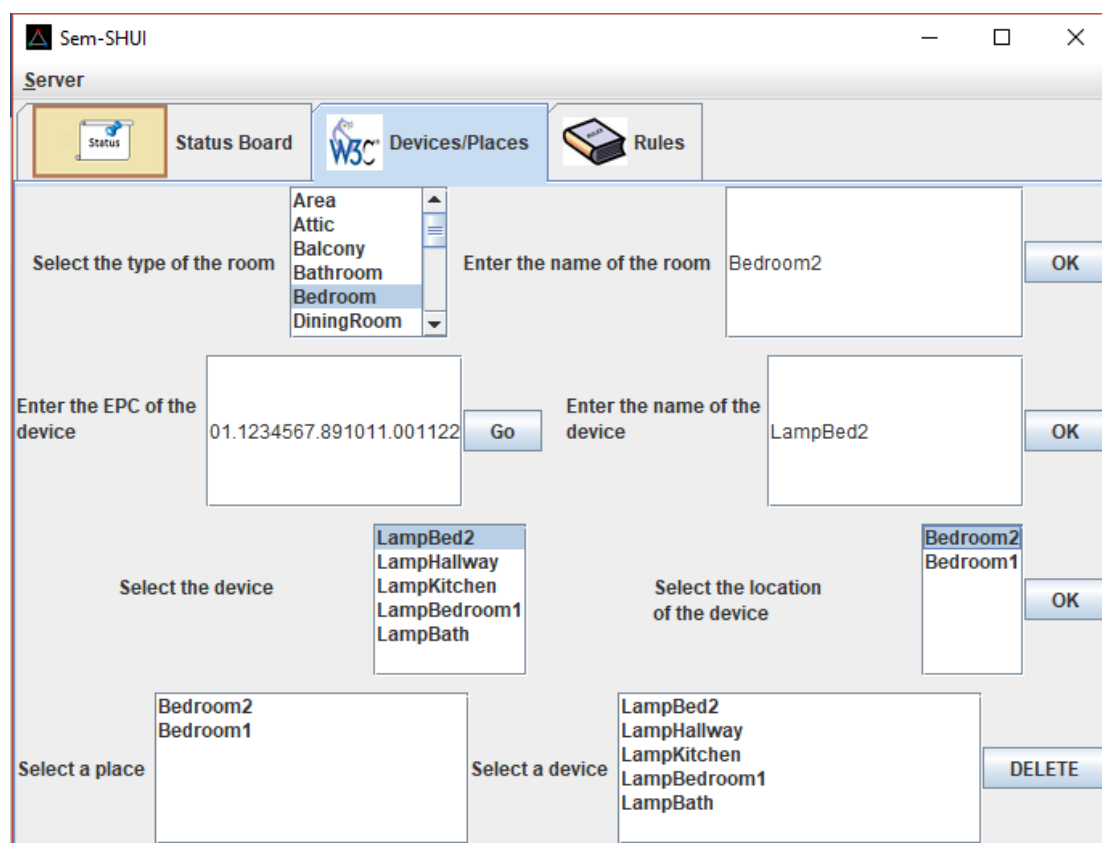
Η παρουσιαζόμενη προσέγγιση έχει ως γνώμονα το χρήστη. Συνεπώς, ο χρήστης πρέπει να έχει τη δυνατότητα να ελέγχει τις συσκευές του και να γράφει πολιτικές σχετικά με τη συμπεριφορά τους μέσω μίας διεπαφής χρήστη. Στην [130] παρουσιάσθηκε μία διεπαφή χρήστη για μη εξειδικευμένους χρήστες στις τεχνολογίες του Σημασιολογικού Ιστού, εξοπλισμένη με αυτές τις δυνατότητες, η οποία ονομάζεται Sem-SHUI (**Semantic Smart Home User Interface**). Παρόλο που στην παρούσα διατριβή εξετάζονται όλα τα περιβάλλοντα του IoT, όχι μόνο το έξυπνο σπίτι, η διεπαφή Sem-SHUI είναι αυτή που εξυπηρετεί το σκοπό της. Αυτή η διεπαφή χρήστη αποτελείται από τρεις καρτέλες: την *Καρτέλα Δημιουργίας Στιγμιότυπων* (ΚΔΣ), την *Καρτέλα Δημιουργίας Κανόνων* (ΚΔΚ), και την *Καρτέλα Κατάστασης Συσκευών* (ΚΚΣ). Η εφαρμογή υλοποιήθηκε χρησιμοποιώντας το πλαίσιο λογισμικού *Jena* [62] καθώς και το πλαίσιο λογισμικού *JADE* [131].

#### 4.1.1 Καρτέλα Δημιουργίας Στιγμιότυπων

Η *Καρτέλα Δημιουργίας Στιγμιότυπων* (*Instance Creation Board*) είναι η καρτέλα που επιτρέπει στο χρήστη να δημιουργήσει στιγμιότυπα των συσκευών του στην πλατφόρμα. Σε σχέση με την [130], προστέθηκε ένας πιο φιλικός τρόπος για να δημιουργούν οι χρήστες τα στιγμιότυπα των συσκευών τους. Αρχικά, ο χρήστης εισάγει τον ηλεκτρονικό κωδικό προϊόντος (EPC) της συσκευής του στο κατάλληλο πεδίο. Η μηχανή αναζήτησης της Sem-SHUI βρίσκει το συγκεκριμένο αντικείμενο που συνοδεύεται από ένα αρχείο OWL το οποίο περιέχει ορισμένες σημαντικές πληροφορίες σχετικά με το προϊόν, όπως οι υπηρεσίες του, οι καταστάσεις, οι εντολές και οι κανόνες του (σε γλώσσα SWRL) σε ορισμένες περιπτώσεις. Οι οντολογίες των συσκευών αντιστοιχίζονται στην οντολογία Core, οπότε η πλατφόρμα ανακαλύπτει αυτόματα τον τύπο της συσκευής. Θα πρέπει να σημειωθεί πως στην παρούσα διατριβή δεν χρησιμοποιήθηκε κάποιο εργαλείο αντιστοίχισης/συγχώνευσης των οντολογιών, όπως για παράδειγμα στην [132], αλλά θεωρήθηκε πως όλες οι συσκευές κάνουν χρήση της οντολογίας Core.

Εάν ο EPC είναι έγκυρος, ο χρήστης πρέπει να εισάγει ένα όνομα για το αντικείμενο. Είναι ανέφικτο για τους χρήστες να θυμούνται τους EPC όλων των προϊόντων τους, δίνοντάς τους έτσι ένα κοινό όνομα (π.χ., LampBed2, Εικόνα 36) είναι πιο φιλικό προς το χρήστη. Επιπλέον, όταν εισάγεται το όνομα, η Sem-SHUI δημιουργεί τον αντίστοιχο SIoT Agent στην πλατφόρμα του SIoT. Ο πράκτορας της συσκευής ανακαλύπτει τις υπηρεσίες από τον κατασκευαστή του αρχείου OWL της αντίστοιχης συσκευής και ξεκινάει τον κύκλο ζωής του (Ενότητα 3.5.1).

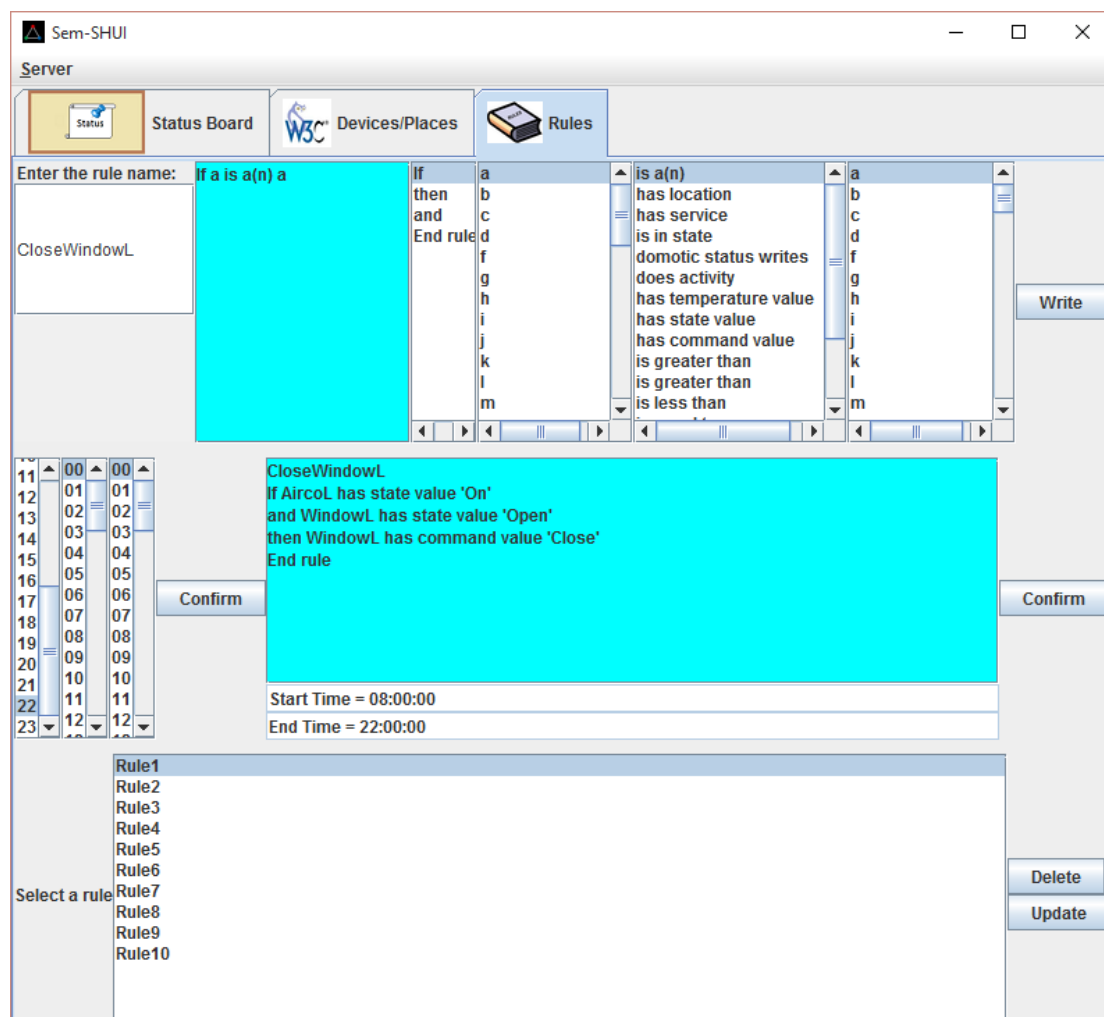
Ο χρήστης έχει επίσης τη δυνατότητα να εισάγει τη θέση μίας συσκευής. Σε πολλές περιπτώσεις ο χρήστης ενδιαφέρεται για τον έλεγχο των περιβαλλοντικών παραμέτρων μίας περιοχής. Για παράδειγμα, ένας χρήστης μπορεί να θέλει να αυξήσει τη θερμοκρασία της κρεβατοκάμαράς του, έτσι εισάγει μία πολιτική ζητώντας από τις συσκευές που εκπέμπουν θερμότητα και βρίσκονται σε αυτό το δωμάτιο να αυξήσουν τη θερμοκρασία του δωματίου. Τέλος, εάν ένα έξυπνο αντικείμενο δεν υφίσταται πια (π.χ., χάλασε), ο κάτοχός του μπορεί να διαγράψει το στιγμιότυπό του (κουμπί DELETE, Εικόνα 36), και κατά συνέπεια τους πράκτορες λογισμικού που αντιστοιχούν σε αυτό. Όλα τα στιγμιότυπα των συσκευών αποθηκεύονται στη βάση triplestore που προσφέρει το πλαίσιο λογισμικού Jena.



Εικόνα 36: Η Καρτέλα Δημιουργίας Στιγμιότυπων της Sem-SHUI

#### 4.1.2 Καρτέλα Δημιουργίας Κανόνων

Η *Καρτέλα Δημιουργίας Κανόνων (Rule Creation Board)* επιτρέπει στο χρήστη να εισάγει δυναμικά σημασιολογικούς κανόνες στο οικοσύστημα του IoT. Συνδυάζει τη λογική της γλώσσας κανόνων Συμβάν-Συνθήκη-Ενέργεια (Event-Condition-Action) [133] με τη λογική των τριπλετών RDF (δηλ., υποκείμενο-κατηγορημα-αντικείμενο). Για παράδειγμα, έστω ότι ο χρήστης θέλει να γράψει έναν κανόνα που θα κλείνει το παράθυρο του καθιστικού όταν το κλιματιστικό είναι αναμμένο. Μετά τη σύνδεση του ενεργοποιητή παραθύρου και της συσκευής κλιματισμού του με την πλατφόρμα SIoT και την ονομασία τους ως "WindowL" και "AircoL" αντιστοίχως, δημιούργησε τον κανόνα "CloseWindowL" (Εικόνα 37). Σύμφωνα με αυτόν τον κανόνα, αν το παράθυρο "WindowL" είναι ανοικτό και το "AircoL" είναι ανοικτό, ο μηχανισμός κίνησης πρέπει να αναγκαστεί να κλείσει το παράθυρο. Όπως φαίνεται στην Εικόνα 37, ο χρήστης πρέπει να δώσει ένα όνομα στον κανόνα του και να καθορίσει την ώρα έναρξης και το χρόνο λήξης του κανόνα, δηλαδή κατά τη διάρκεια ποιας χρονικής περιόδου ο κανόνας του θα είναι ενεργός.



Εικόνα 37: Η Καρτέλα Δημιουργίας Κανόνων της Sem-SHUI

Ένας άλλος τρόπος να γραφτεί ο παραπάνω κανόνας βάσει της περιοχής που γενικεύει την ιδέα του "CloseWindowL" παρουσιάζεται σε μορφή SWRL στον Πίνακα 3. Σε αυτή την περίπτωση, ο χρήστης θα πρέπει να έχει εισαγάγει το χώρο του σαλονιού του και να έχει προκαθορίσει την τοποθεσία των έξυπνων συσκευών του. Θα πρέπει να σημειωθεί ότι εάν υπάρχουν περισσότεροι ενεργοποιητές παραθύρων στο σαλόνι θα δοθεί και σε αυτούς εντολή να κλείσουν. Η διεπαφή Sem-SHUI έχει σχεδιασθεί για να αποθηκεύει στο αποθετήριο κανόνων μόνο τους κανόνες που είναι συνεπείς. Εάν κανένας κανόνας δεν εκφρασθεί σωστά ή δεν έχει καθοριστεί μία υποχρεωτική τιμή (π.χ., ο χρήστης δεν έχει ονομάσει τον κανόνα του), ο χρήστης καλείται να τον ξαναγράψει.

Μετά την ολοκλήρωση του κανόνα, η διεπαφή Sem-SHUI αναλύει την περιοχή "RulesArea", η οποία περιέχει τον κανόνα σε μορφή αναγνώσιμη από τον άνθρωπο και μεταφράζεται σε στιγμιότυπα OWL σύμφωνα με την οντολογία Rule (Ενότητα 3.4.4). Συγκεκριμένα, η διεπαφή Sem-SHUI δημιουργεί ένα νέο στιγμιότυπο κανόνα. Με βάση το προηγούμενο παράδειγμα, θα δημιουργηθεί το στιγμιότυπο Δράστη "WindowLCloseWindowL". Ο Πράκτορας Χρήστη που είναι ενσωματωμένος στη διεπαφή Sem-SHUI, στέλνει ένα μήνυμα στον αντίστοιχο SIoT Agent του "WindowL", ενεργοποιώντας τη συμπεριφορά ανίχνευσης σύγκρουσης. Αφού ολοκληρώσει την εργασία του, το *ConformityReasoner* ενημερώνει τον User Agent, μέσω του *CommunicationHandler* για τα αποτελέσματα. Στην επόμενη ενότητα επεξηγείται λεπτομερώς η όλη διαδικασία ανίχνευσης συγκρούσεων. Τέλος, ο χρήστης μπορεί να διαγράψει προηγούμενους κανόνες ή να τους ενημερώσει, πατώντας τα κουμπιά Delete και Update αντίστοιχως.

Πίνακας 3: Κανόνας με βάση την τοποθεσία των αντικειμένων

Όνομα Κανόνα	Κώδικας Κανόνα (SWRL)
CloseWindowL	$\text{LivingRoom}(\text{?L}) \wedge \text{WindowActuator}(\text{?W}) \wedge \text{Aircondition}(\text{?A}) \wedge$ $\text{hasLocation}(\text{?W}, \text{?L}) \wedge \text{hasLocation}(\text{?A}, \text{?L}) \wedge$ $\text{hasStateValue}(\text{?W}, \text{"Open"}) \wedge \text{hasStateValue}(\text{?A}, \text{"On"})$ $\rightarrow \text{hasCommandValue}(\text{?W}, \text{"Off"})$

### 4.1.3 Καρτέλα Κατάστασης Συσκευών

Η τρίτη καρτέλα της διεπαφής Sem-SHUI είναι η *Καρτέλα Κατάστασης Συσκευών* (Device Status Board). Η ΚΚΣ είναι ο πίνακας ελέγχου της διεπαφής Sem-SHUI. Ο χρήστης λαμβάνει μηνύματα και ειδοποιήσεις από τους πράκτορες των έξυπνων αντικειμένων που είναι συνδεδεμένα στην πλατφόρμα SIoT, τα οποία σχετίζονται με την αλλαγή μίας κατάστασης. Όλες οι καταστάσεις των συσκευών του και οι εντολές, στις οποίες ακούν, παρατίθενται σε αυτόν τον πίνακα. Μέσω της ΚΚΣ ο χρήστης μπορεί να αλλάξει την κατάσταση ενός έξυπνου αντικειμένου χρησιμοποιώντας μία



άμεση εντολή. Η εντολή προωθείται από τον User Agent στον κατάλληλο πράκτορα συσκευής και στη συνέχεια αυτός στέλνει την εντολή στη συσκευή που αντιπροσωπεύει. Σε αντίθεση με τους πράκτορες συσκευών, οι πράκτορες που αντιπροσωπεύουν αισθητήρες δεν μπορούν να λάβουν εντολές. Η μόνη ευθύνη τους είναι να ενημερώνουν τους συνδρομητές τους και το χρήστη, μέσω Pub-Sub, σχετικά με αλλαγές που συμβαίνουν στις παραμέτρους που παρατηρούν. Αυτή η υπηρεσία περιλαμβάνεται στον πίνακα "DO", που βρίσκεται στο πάνω μέρος της καρτέλας ΚΚΣ (Εικόνα 38).

Ένα άλλο σημαντικό χαρακτηριστικό αυτής της καρτέλας είναι ο πίνακας "NEED", όπου λαμβάνει χώρα ο μηχανισμός ανακάλυψης των υπηρεσιών των SIoT Agents. Μέσω της συγκεκριμένης διεπαφής ο χρήστης μπορεί να ορίζει τις ανάγκες του. Όπως, περιγράφεται στην Ενότητα 3.5.1, οι πράκτορες λογισμικού είναι υπεύθυνοι για την ανακάλυψη των αντικειμένων/υπηρεσιών που είναι σε θέση να ικανοποιήσουν τις ανάγκες του χρήστη. Για παράδειγμα, εάν ο χρήστης χρειάζεται φωτισμό στο σαλόνι, ο Πράκτορας Χρήστη δημιουργεί δυναμικά ένα ερώτημα SPARQL που θα επιστρέψει σε όλες τις συσκευές/εφαρμογές με τις οποίες είναι φίλος και είναι εξοπλισμένες με αυτή τη λειτουργία (Εικόνα 39). Αυτό το ερώτημα βασίζεται στις οντολογίες Core, SSOR, και SONS. Στη συνέχεια, ο χρήστης μπορεί να επιλέξει τη συσκευή που θέλει να εκτελέσει αυτή τη λειτουργία. Η Εικόνα 39 δείχνει το ερώτημα SPARQL για εύρεση συσκευών που προσφέρουν φως και βρίσκονται στο υπνοδωμάτιο, γραμμένο στην καρτέλα ερωτημάτων SPARQL του protégé.

The screenshot shows the Sem-SHUI interface with the following components:

- Server Window:** Contains tabs for Status Board, Devices/Places, and Rules.
- DO Section:** A table for defining actions.
 

Agent	DO	Turn on	Lamp1	00	00	00
		Turn on 50%	Door1	01	01	01
		Turn off	WindowL	02	02	02
		Open	TV1	03	03	03
		Close	LampBedroom1	04	04	04
- NEED Section:** A table for defining needs.
 

Agent	NEED	Light	Bathroom	14	00	00
Agent NEED Light in Bedroom1 at 20:00:00		Cold	Bedroom1	15	01	01
		Heat	Bedroom2	16	02	02
			Kitchen	17	03	03
			Livingroom	18	04	04
			Toilet	19	05	05
			Hallway	20	06	06
			LivingroomEntrance	21	07	07
			KitchenEntrance	22	08	08
			BathroomEntrance	23	09	09
- Log Section:**

```
UserAgent: [DO] Turn on TV1 at 19:55:00
Door1Agent: Opened at 19:59:15
UserAgent: [NEED] Light in Bedroom1 at 20:00:00
```

Εικόνα 38: Η Καρτέλα Κατάστασης Συσκευών της διεπαφής Sem-SHUI

```

SPARQL query:
PREFIX smart:<http://www.icbnet.ntua.gr/smarthome/ontologies/smarthome#>
PREFIX ssor:<http://consert.teipir.gr/research/results/consert-in-c-computing/ontologies/ssor-ont.owl#>
PREFIX sons:<http://consert.teipir.gr/research/results/consert-in-c-computing/ontologies/sons-ont.owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?need ?object
WHERE {
  ?need a sons:Need;
        a smart:Light.
  ?area a smart:Bedroom;
        hasName:Bedroom1.
  ?object a ssor:SocialFriend;
          sons:offers ?service;
          smart:hasLocation ?area.
  ?need sons:isSatisfiedBy ?service.}

```

Εικόνα 39: Ερώτημα SPARQL για την εύρεση μίας συσκευής που προσφέρει φως στο δωμάτιο Bedroom1.

Θα πρέπει να σημειωθεί ότι οι εντολές "DO" και "NEED" θεωρούνται πιο ισχυρές από τους κανόνες (δηλαδή έχουν προτεραιότητα). Για παράδειγμα, εάν ο κανόνας "WindowCloseL" είναι ενεργός και ο χρήστης παρακινεί τον ενεργοποιητή "WindowL" να ανοίξει, ο πράκτορας θα εκτελέσει αυτή την εντολή, παρά το γεγονός ότι ο κανόνας έχει ως έξοδο την ακριβώς αντίθετη εντολή. Αυτή η ιεραρχία δίνει ένα πλεονέκτημα στις άμεσες εντολές έναντι των έμμεσων εντολών, δηλαδή των κανόνων.

#### 4.1.4 Μηχανισμός ανίχνευσης συγκρούσεων κανόνων

Το πλεονέκτημα της χρήσης σημασιολογικών κανόνων στο περιβάλλον του Διαδικτύου των Πραγμάτων είναι ότι οι οντολογίες ενσωματώνουν τη γνώση του οικοσυστήματος (π.χ., τις σχέσεις μεταξύ των οντοτήτων του). Σε αυτή την ενότητα περιγράφεται λεπτομερώς ο εκτός σύνδεσης μηχανισμός ανίχνευσης συγκρούσεων. Στην πρώτη ενότητα παρουσιάζεται ολόκληρη η λογική των σημασιολογικών κανόνων, ενώ στη δεύτερη δίνεται έμφαση στις λεπτομέρειες της εφαρμογής της στην προτεινόμενη πλατφόρμα του SIoT.

##### 4.1.4.1 Λογική πρώτου βαθμού για τους σημασιολογικούς κανόνες

Οι σημασιολογικοί κανόνες συνίστανται από ένα σύνολο αληθών *ατόμων* [134]. Σύμφωνα με την [135] μία οντολογία μπορεί να χωρισθεί σε δομικά στοιχεία, που ονομάζονται *άτομα* (*atoms*) ή *ατομικές εκφράσεις* (*atomic formula*), τα οποία δεν μπορούν να χωρισθούν περαιτέρω σε δομικά στοιχεία. Όπως ορίζεται στον προτασιακό λογισμό, μία φράση είναι έγκυρη εάν όλα τα *άτομα* του πρώτου μέρους, που ονομάζονται επίσης σώμα της φράσης, είναι αληθή. Επομένως, εάν το σώμα ενός

κανόνα είναι έγκυρο τότε ενεργοποιείται η συνέπεια, που ονομάζεται επίσης κεφαλή της φράσης. Ένας σημασιολογικός κανόνας μπορεί να εκφρασθεί ως:

$$p_1(arg1, arg2, \dots, argn) \wedge p_2(arg1, arg2, \dots, argn) \dots \wedge p_n(arg1, arg2, \dots, argn) \rightarrow p(arg1, arg2, \dots, argn) \quad (3.1)$$

όπου ο όρος  $p_i(arg1, arg2, \dots, argn)$  υποδηλώνει ένα άτομο. Ο χαρακτήρας  $p$  είναι το σύμβολο που αντιπροσωπεύει μία λογική δήλωση και τα  $arg1, arg2, \dots, argn$  είναι τα επιχειρήματα της φράσης. Το σύμβολο  $p$  μπορεί να αντιπροσωπεύει κλάσεις OWL, ιδιότητες αντικειμένου ή ιδιότητες δεδομένων. Τα επιχειρήματα μπορεί να είναι άτομα OWL ή τιμές δεδομένων ή μεταβλητές που τις υποδηλώνουν. Ο Πίνακας 4 παρουσιάζει όλους τους διαφορετικούς τύπους ατόμων. Θα πρέπει να σημειωθεί ότι τα άτομα διακύμανσης δεδομένων δεν είναι διαθέσιμα στην ΚΔΚ της διεπαφής Sem-SHUI, καθώς είναι πολύ περίπλοκα για τους χρήστες που δεν είναι προγραμματιστές.

Πίνακας 4: Διαφορετικοί τύποι ατόμου

Τύπος Ατόμου	Παράδειγμα σε SWRL
Κλάση	Lamp(?l)
Ιδιότητα Αντικειμένου	hasLocation(?l, ?area)
Ιδιότητα Δεδομένων	hasStateValue(?l, "On")
Διαφορετικά Στιγμιότυπα	differentFrom(?lamp1, ?lamp2)
Ίδια Στιγμιότυπα	sameAs(?lamp1, ?lamp2)
Ενσωματωμένες Συναρτήσεις	greaterThan(?value, 25)
Εύρος Δεδομένων	xsd:int(?x)

Στο προτεινόμενο οικοσύστημα για το SIoT, δύο κανόνες μπορεί να είναι αντιφατικοί εάν οι συνέπειές τους οδηγούν σε αντίθετο αποτέλεσμα. Για παράδειγμα, έστω ότι υπάρχουν δύο κανόνες, ο rule1 και ο rule2, όπου ο rule1 αναγκάζει μία λάμπα να ενεργοποιηθεί και ο rule2 την αναγκάζει να απενεργοποιηθεί. Ως αποτέλεσμα, αν υπάρχει χρονικό στιγμιότυπο όπου δύο ή περισσότεροι κανόνες είναι ενεργοί και δίνουν ταυτόχρονα εντολή στο ίδιο αντικείμενο να έχει διαφορετικές τιμές κατάστασης, τότε υπάρχει σύγκρουση μεταξύ τους. Επιπλέον, παρόλο που δύο κανόνες μπορεί να έχουν αντίθετα αποτελέσματα,

αν τα χρονικά διαστήματα κατά τη διάρκεια των οποίων είναι ενεργοί (δηλ., το χρονικό διάστημα που αντιστοιχεί στον καθένα μεταξύ του χρόνου έναρξης και του χρόνου λήξης) δεν είναι αλληλεπικαλυπτόμενα τότε δεν θεωρούνται ως αντικρουόμενοι. Η προτεινόμενη προσέγγιση για την εύρεση των στιγμών όπου υπάρχουν δύο αντιφατικοί κανόνες, υιοθετεί τη λογική της εξασφάλισης ότι οι κανόνες δεν είναι αντιφατικοί, δηλαδή δεν υπάρχει καμία πιθανότητα δύο κανόνες να συγκρούονται μεταξύ τους. Η προσέγγιση αυτή ονομάζεται “Αρχή μη-σύγκρουσης”.

Ένας κανόνας, λοιπόν, αποτελείται από ένα σύνολο ατόμων που μπορεί να αποτυπωθεί ως μεταβλητές  $X = \{X_1, X_2, \dots, X_n\}$  και κάθε μία μεταβλητή  $X_i$  μπορεί να πάρει τιμές από ένα πεδίο ορισμού  $D_i$ . Για λόγους απλότητας, έστω ότι οι μεταβλητές μπορούν να πάρουν μόνο δυαδικές τιμές  $D_i = \{0, 1\}$ . Πιο συγκεκριμένα, θεωρείται ότι ένα έξυπνο αντικείμενο μπορεί να έχει δύο τιμές κατάστασης, “On” και “Off”. Δηλαδή, η συσκευή  $X_i^0$  βρίσκεται σε κατάσταση “Off” και η συσκευή  $X_i^1$  σε κατάσταση “On”.

Η “Αρχή μη-σύγκρουσης” ορίζεται ως εξής: Υποθέτοντας πως υπάρχουν δύο κανόνες,  $rule_i$  and  $rule_j$ , αν οι συνέπειες τους είναι αντικρουόμενες και υπάρχει μία χρονική στιγμή όπου είναι και οι δύο ενεργοί, είναι μη-αντικρουόμενοι αν και μόνο αν υπάρχει μία μεταβλητή στο σώμα τους που να έχει διαφορετική τιμή σε ένα πεδίο ορισμού.

$$\begin{aligned} rule_i &= \{X_1^0, \dots, X_n^0\} \\ rule_j &= \{X_1^1, \dots, X_k^0\} \end{aligned} \quad (3.2)$$

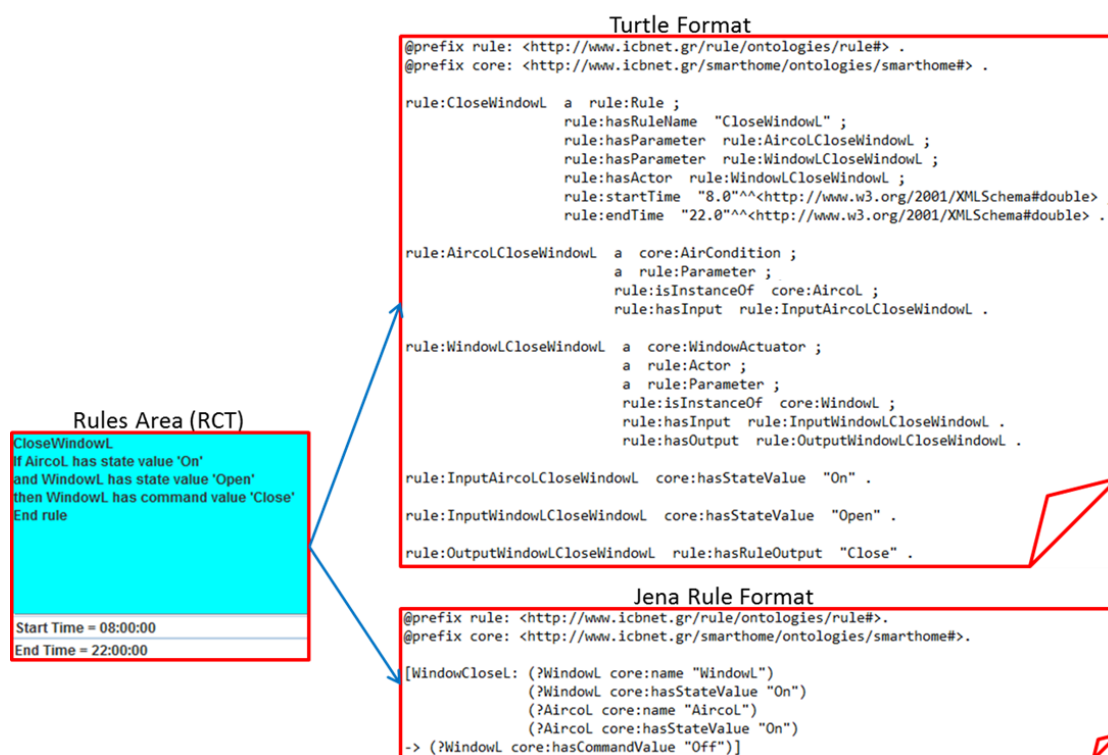
Ο κανόνας  $rule_i$  αποτελείται από ένα σύνολο  $n$  μεταβλητών και ο κανόνας  $rule_j$  από ένα σύνολο  $k$  μεταβλητών. Στη συγκεκριμένη περίπτωση, η μεταβλητή  $X_1$  εξασφαλίζει πως δεν υπάρχει κάποια σύγκρουση μεταξύ τους. Αυτό προκύπτει εξαιτίας του ότι μία συσκευή δεν μπορεί να έχει δύο διαφορετικές τιμές κατάστασης την ίδια χρονική στιγμή,  $P(X_1^0) \wedge P(X_1^1) = \emptyset$ , δηλαδή δεν τηρείται η αρχή ικανοποιησιμότητας (Ενότητα 2.2.3). Συνεπώς, το σύνολο τιμών των μεταβλητών που ικανοποιούν ταυτόχρονα όλες τις συνθήκες των δύο κανόνων είναι ίσο με το άδειο σύνολο  $\{\}$ . Σε όλες τις άλλες περιπτώσεις, μπορεί να υπάρξει μία κατάσταση περιβάλλοντος όπου θα ενεργοποιηθούν και οι δύο κανόνες. Τέλος, θα πρέπει να σημειωθεί το γεγονός ότι σημασιολογικοί κανόνες δεν χρησιμοποιούν μη μονοτονική λογική, δηλαδή οι κανόνες δεν μπορούν να διαγράψουν την προηγούμενη γνώση μίας οντολογίας.

#### 4.1.4.2 Ροή εργασιών αποθήκευσης κανόνων

Μετά τη δημιουργία ενός κανόνα στην ΚΔΚ της διεπαφής Sem-SHUI, ο χρήστης ενεργοποιεί τη ροή εργασιών αποθήκευσης κανόνων, πατώντας το κουμπί

επιβεβαίωσης (Confirm). Προκειμένου να επιτευχθεί η διαδικασία ανίχνευσης των συγκρούσεων, υλοποιήθηκαν δύο συντακτικοί αναλυτές (parsers), ο *RuleTurtleTranslator* και ο *RuleTransformationParser*.

Το καθήκον του *RuleTranslator* είναι να αναλύσει τον τομέα κανόνων της ΚΔΚ και να μεταφράσει το περιεχόμενο σε ένα αρχείο turtle, βασισμένο στην οντολογία Rule και εκμεταλλευόμενο την οντολογία κορμού, το οποίο θα περιέχει το στιγμιότυπο αυτού του συγκεκριμένου κανόνα και όλων των πληροφοριών που είναι σχετικές με τον κανόνα και είναι υποχρεωτικές για το *ConformityReasoner*. Ένα απλό παράδειγμα επιδεικνύεται στην Εικόνα 40. Η ιδέα πίσω από την αναπαράσταση του κανόνα είναι η προσομοίωση μίας παγκόσμιας κατάστασης στην οποία όλοι οι κανόνες είναι ενεργοί, δηλαδή όλα τα αξιώματά τους ικανοποιούνται. Όπως αναφέρθηκε προηγουμένως, κάθε κανόνας πρέπει να έχει ένα όνομα, ένα χρόνο έναρξης, ένα χρόνο λήξης, τουλάχιστον μία παράμετρο και τουλάχιστον έναν δράστη. Τα στιγμιότυπα του δράστη και της παραμέτρου συνδέονται με τον κανόνα στον οποίο αναφέρονται (στιγμιότυπα κανόνων) και φυσικά με το στιγμιότυπο του φυσικού αντικειμένου.



Εικόνα 40: Παράδειγμα μετατροπής ενός κανόνα από τη διεπαφή Sem-SHUI σε οντολογικά στιγμιότυπα, και στη συνέχεια σε σημασιολογικό κανόνα.

Μετά την οντολογική αναπαράσταση ενός του κανόνα, το αρχείο turtle διαβιβάζεται στον User Agent του Sem-SHUI και αποθηκεύεται τοπικά σε ένα αρχείο, στο οποίο έχει πρόσβαση το *ConformityReasoner* του δράστη. Ο User Agent στέλνει

ένα μήνυμα ειδοποίησης στον αντίστοιχο SIoT Agent, το οποίο ξυπνά τη συμπεριφορά ανίχνευσης συγκρούσεων του *ConformityReasoner*. Αυτός με τη σειρά του, διαβάζει το νέο αρχείο turtle και εκτελεί αμέσως ένα ερώτημα SPARQL που εξυπηρετεί την “Αρχή μη-σύγκρουσης” (Εικόνα 41). Αν υπάρχουν δύο κανόνες που έχουν ως δράστη το στιγμιότυπο του ίδιου αντικειμένου και το αποτέλεσμα (έξοδος κανόνα) του κάθε κανόνα αναφέρεται στον ίδιο τύπο εντολής, αλλά δίνει μία διαφορετική τιμή κατάστασης (π.χ., μείωσε την ένταση της τηλεόρασης και αύξησε την ένταση της τηλεόρασης), ενώ υπάρχει ένα χρονικό διάστημα όπου αυτοί οι δύο κανόνες είναι ενεργοί, τότε μπορεί να είναι συγκρουόμενοι. Το ερώτημα SPARQL, λοιπόν, ελέγχει όλες τις παραμέτρους του και αν υπάρχει ένα στιγμιότυπο αντικειμένου με βάση τον κανόνα που περιέχεται και στα δύο σώματα των κανόνων ενώ παράλληλα βρίσκεται τον ίδιο τύπο κατάστασης αλλά με διαφορετικές τιμές κατάστασης, τότε δεν είναι συγκρουόμενες. Τέλος, η ταξινόμηση της κατάστασης και της δράσης εκτελείται χρησιμοποιώντας το συμπερασματικό λογισμό στην οντολογία Core. Για παράδειγμα, μία κατάσταση που έχει τιμή κατάστασης "On" ή "Off" ταξινομείται ως OnOffState.

```

PREFIX core: <http://www.icbnet.gr/smarthome/ontologies/smarthome#>
PREFIX rule: <http://www.icbnet.gr/rule/ontologies/rule#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?rule1 ?rule2
WHERE {
    ?rule1 a rule:Rule;
           rule:hasActor ?actor1;
           rule:startTime ?start1;
           rule:endTime ?end1;
           rule:hasParameter ?param1.
    ?rule2 a rule:Rule;
           rule:hasActor ?actor2;
           rule:hasParameter ?param2;
           rule:startTime ?start2;
           rule:endTime ?end2.
    ?actor1 rule:hasOutput ?output1;
           rule:isInstanceOf ?object1.
    ?actor2 rule:hasOutput ?output2;
           rule:isInstanceOf ?object2.
    ?output1 a ?action1;
            rule:hasRuleOutput ?ruleOutput1.
    ?output2 a ?action2;
            rule:hasRuleOutput ?ruleOutput2.
    ?param1 a ?class1;
            rule:hasInput ?input1.
    ?input1 a ?state1;
            core:hasStateValue ?value1.
    ?param2 a ?class2;
            rule:hasInput ?input2.
    ?input2 a ?state2;
            core:hasStateValue ?value2.
    FILTER (?object1 = ?object2 && ?class1 = ?class2 && ?value1 != ?value2
            && ?state1 = ?state2 && ?ruleOutput1 != ?ruleOutput2 &&
            ?action1 = ?action2 && ?start1 >= ?start2 && ?start1 <= ?end2)}}

```

Εικόνα 41: Ερώτημα SPARQL για την εύρεση των μη αντικρουόμενων κανόνων, βασισμένο στην “Αρχή μη-σύγκρουσης”.

Ο Πίνακας 5 δείχνει ένα παράδειγμα δύο κανόνων που ενώ φαινομενικά συγκρούονται, τελικά δεν είναι αντιφατικοί. Ο κανόνας rule1 έχει ως παραμέτρους



του τα στιγμιότυπα *sensor1*, και *tv1* και ως δράστη το στιγμιότυπο *airco1*. Η τιμή της θερμοκρασίας που παρατηρεί ο *sensor1* είναι 28°C. Η έξοδος του *rule1* είναι να αλλάξει η τιμή της θερμοκρασίας του κλιματιστικού *airco1* σε 25°C. Ο *rule2* έχει ως παραμέτρους του τα στιγμιότυπα *sensor1*, *lamp1* και ως δράστη το στιγμιότυπο *airco1*. Σε αυτή την περίπτωση, η τιμή της θερμοκρασίας που παρατηρεί ο *sensor1* πρέπει να ισούται με 20°C, για να ενεργοποιηθεί ο κανόνας. Η έξοδος του *rule2* είναι να αλλάξει η τιμή της θερμοκρασίας του κλιματιστικού *airco1* σε 22°C. Και οι δύο κανόνες έχουν χρόνο έναρξης 8:00 π.μ. και ώρα λήξης 10:00 μ.μ. Αυτοί οι δύο κανόνες έχουν διαφορετικό αποτέλεσμα σχετικά με το ίδιο αντικείμενο, επομένως, ενδεχομένως να υπάρχει σύγκρουση. Ωστόσο, η θερμοκρασία στην περιοχή που βρίσκεται ο αισθητήρας *sensor1* δεν μπορεί να έχει ταυτόχρονα 28°C και 20°C. Ως αποτέλεσμα, δεν υπάρχει πιθανότητα ο *rule1* να έρχεται σε σύγκρουση με τον *rule2*.

**Πίνακας 5: Παράδειγμα μη αντικρουόμενων κανόνων**

Όνομα κανόνα	Μορφή SWRL	Active time
<b>rule1</b>	TV(tv1) ^ hasStateValue(tv1, 'On') ^ Sensor(sensor1) ^ hasObservationValue(sensor1, 28) -> hasCommandValue(airco1, 25)	8:00:00 – 22:00:00
<b>rule2</b>	Lamp(lamp1) ^ hasStateValue(lamp1, 'On') ^ Sensor(sensor1) ^ hasObservationValue(sensor1, 20) -> hasCommandValue(airco1, 22)	8:00:00 – 22:00:00

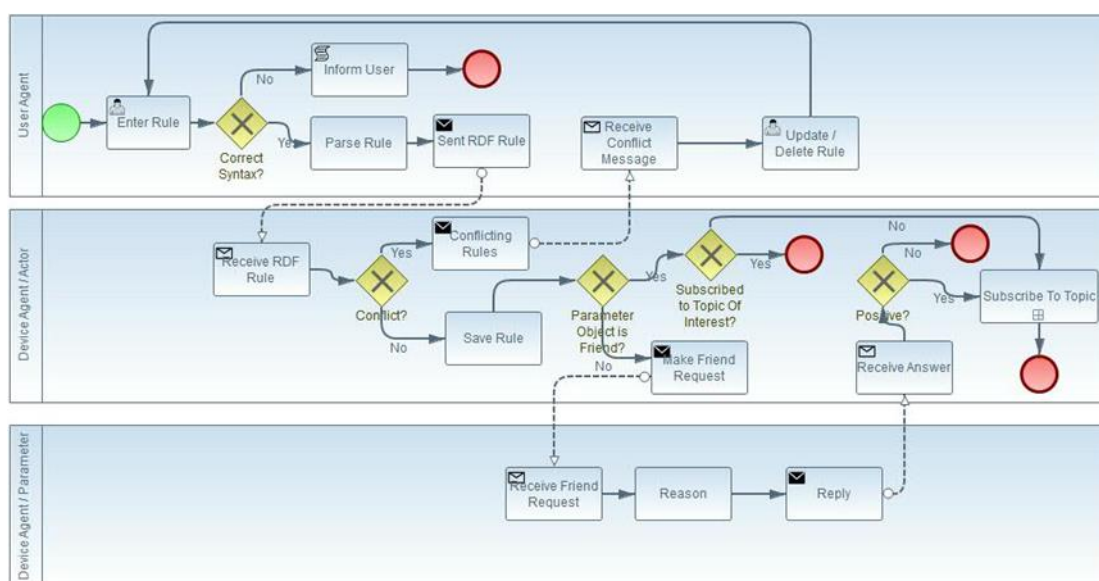
Το στοιχείο *ConformityReasoner* πρέπει να γνωρίζει όλους τους κανόνες που ισχύουν για τη συσκευή που εκπροσωπεί στην πλατφόρμα *SlIoT*, κρατώντας μία λίστα. Εάν η λίστα αποτελεσμάτων του ερωτήματος SPARQL δεν περιέχει το νέο κανόνα, αυτό σημαίνει ότι βρίσκεται σε σύγκρουση με έναν άλλο κανόνα. Συγκρίνοντας τον κατάλογο που προκύπτει με τον προηγούμενο, ανακαλύπτει ποιοι κανόνες δεν υπάρχουν και στα δύο. Αυτοί οι κανόνες είναι οι αντικρουόμενοι.

Όταν ολοκληρωθεί η ροή εργασιών ανίχνευσης συγκρούσεων, ο *SlIoT Agent* ενημερώνει τον *User Agent* της διεπαφής *Sem-SHUI* για τα αποτελέσματα. Εάν υπάρχει σύγκρουση, ο χρήστης καλείται από το *Sem-SHUI* να διαγράψει έναν από τους αντικρουόμενους κανόνες ή να αλλάξει το νέο κανόνα, για παράδειγμα εισάγει ορισμένες νέες παραμέτρους. Διαφορετικά, ο κανόνας μετατρέπεται σε μορφή *Jena* (Εικόνα 40), από τον *RuleTransformationParser*. Θα πρέπει να σημειωθεί ότι ο κανόνας σε μορφή *Jena* δεν περιέχει τις τιμές χρόνου έναρξης και λήξης. Αυτές οι τιμές είναι αποθηκευμένες μαζί με το όνομα των κανόνων σε μία βάση δεδομένων *NoSQL* (συγκεκριμένα *MongoDB*) που αποτελείται από αρχεία *json*. Το *ConformityReasoner* αποθηκεύει το αρχείο κανόνα σε μορφή *Jena* στη βάση δεδομένων κανόνων του *SlIoT*

Agent, ο οποίος ελέγχει το έξυπνο αντικείμενο που είναι ο δράστης σε αυτόν το συγκεκριμένο κανόνα. Η βάση δεδομένων κανόνων του πράκτορα αποτελείται από όλους τους ισχύοντες κανόνες που αφορούν τη συγκεκριμένη συσκευή. Στην προτεινόμενη SIoT πλατφόρμα οι κανόνες που είναι εγγεγραμμένοι από τον κατασκευαστή της συσκευής (αναπαρίστανται σε μορφή SWRL στο αρχείο OWL της συσκευής) συνεργάζονται με εκείνους που εισάγονται από τους χρήστες και διατηρούνται στην ίδια βάση αποθήκευσης.

#### 4.1.5 Μηχανισμός δημιουργίας στόχων

Η δυναμική εισαγωγή κανόνων από τους χρήστες, μέσω της διεπαφής Sem-SHUI, οδηγεί στη δημιουργία νέων στόχων. Επειδή το προτεινόμενο SIoT οικοσύστημα είναι αποκεντριοποιημένο και κατανεμημένο, για την επιβολή ενός κανόνα που θα έχει ως αποτέλεσμα τη μεταβολή της κατάστασης ενός έξυπνου αντικειμένου, θα πρέπει ο πράκτορας λογισμικού του να είναι εγγεγραμμένος στα θέματα που σχετίζονται με τις παραμέτρους του κανόνα. Πιο συγκεκριμένα, ο πράκτορας λογισμικού του Actor θα πρέπει να είναι φίλος με όλες τις συσκευές που συμμετέχουν ως παράμετροι στο σώμα του κανόνα και να παρακολουθεί την κατάσταση στην οποία βρίσκονται. Για παράδειγμα, ο κανόνας *rule1* (Πίνακας 5) έχει ως δράστη το *airco1* και ως παραμέτρους την *tv1* και το *sensor1*. Για να δώσει εντολή στο *airco1* (δηλ., να αποκτήσει θερμοκρασία λειτουργίας ίση με 25) ο πράκτοράς του θα πρέπει να γνωρίζει την τιμή (δηλ., θερμοκρασία) που παρατηρεί ο *sensor1* και αν η *tv1* είναι ανοικτή. Συνεπώς, όχι μόνο θα πρέπει να είναι φίλος με τους αντίστοιχους SIoT Agents αλλά και να είναι συνδρομητής στα θέματα που σχετίζονται με τις συγκεκριμένες λειτουργίες/καταστάσεις των συσκευών.



Εικόνα 42: BPMN διάγραμμα περιγραφής της ροής εργασιών κατά την εισαγωγή νέου κανόνα.



Η Εικόνα 42 παρουσιάζει τη ροή εργασιών κατά την εισαγωγή νέου κανόνα σε μορφή BPMN. Όπως περιγράφεται και στην προηγούμενη ενότητα, ο χρήστης εισάγει αρχικά τον κανόνα του στην ΚΔΚ και ο UserAgent ελέγχει αν ο κανόνας είναι συντακτικά ορθός. Αν είναι έγκυρος ο κανόνας, τον μετατρέπει σε τριπλέτες και τον στέλνει στον πράκτορα λογισμικού που αποτελεί το δράστη του κανόνα, ενώ αν δεν είναι έγκυρος ενημερώνει το χρήστη. Όταν ο DeviceAgent λάβει το μήνυμα, ελέγχει αν υπάρχει σύγκρουση με άλλον κανόνα. Σε περίπτωση που ανιχνευθεί κάποια σύγκρουση ενημερώνει τον UserAgent. Αν ο κανόνας είναι αντικρουόμενος, ο DeviceAgent τον εισάγει στο αρχείο με τους υπόλοιπους κανόνες αυτοματισμού, και στη συνέχεια εξετάζει αν είναι φίλος με όλες τις παραμέτρους του κανόνα και αν είναι συνδρομητής στα επιθυμητά θέματα. Αυτό επιτυγχάνεται μέσω ενός ερωτήματος SPARQL (Εικόνα 43).

```

SPARQL query:
PREFIX ssor: <http://consert.puas.gr/research/results/consert-in-c-computing/ontologies/ssor-ont.owl>
PREFIX sons: <http://consert.puas.gr/research/results/consert-in-c-computing/ontologies/sons-ont.owl>
PREFIX rule: <http://www.icbnet.gr/rule/ontologies/rule#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?object ?topic
WHERE {
    ?rule rule:hasParameter ?objectInstance;
           rule:hasRuleName ?ruleName.
    ?objectInstance rule:isInstanceOf ?object;
                   rule:hasInput ?state.
    ?state sons:describedBy ?service.
    ?service sons:isRelatedTo ?topic.
    ?me ssor:hasFriend ?object.
    ?me sons:subscribesTo ?topic.
    FILTER regex(?ruleName, "ruleName") }
    
```

Εικόνα 43: Ερώτημα SPARQL του πράκτορα δράστη για την επιβεβαίωση πως οι παράμετροι ενός συγκεκριμένου κανόνα είναι φίλοι του και για την εξέταση αν είναι συνδρομητής στα απαραίτητα θέματα.

```

SPARQL query:
PREFIX core: <http://www.icbnet.gr/smarthome/ontologies/smarthome#>
PREFIX sons: <http://consert.teipir.gr/research/results/consert-in-c-computing/ontologies/sons-ont.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?uri
WHERE {
    ?state core:possibleValues ?value;
           sons:describedBy ?service.
    ?service sons:isRelatedTo ?topic.
    ?topic sons:hasTopicURI ?uri.
    FILTER regex(?value, "valueName")}
    
```

Εικόνα 44: Ερώτημα SPARQL του πράκτορα της παραμέτρου του κανόνα για την εύρεση του URI του topic για το οποίο ενδιαφέρεται ο πράκτορας λογισμικού του δράστη.

Στην περίπτωση που ο πράκτορας του δράστη δεν είναι φίλος με τις παραμέτρους του κανόνα, στέλνει αίτημα φιλίας στους πράκτορες που τις εκπροσωπούν. Εφόσον ο πράκτορας λογισμικού της παραμέτρου αποδεχθεί το αίτημα τότε ο πράκτορας λογισμικού του δράστη του στέλνει ένα μήνυμα, έχοντας

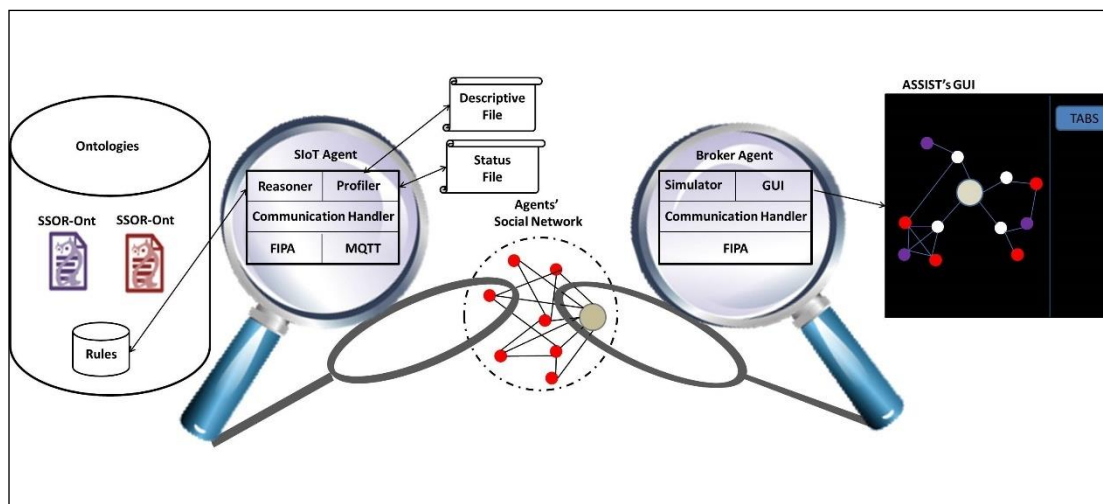
ως στόχο να ενημερωθεί για το URI του θέματος (*hasTopicURI*) στο οποίο τον ενδιαφέρει να γίνει συνδρομητής. Αυτό επιτυγχάνεται και πάλι μέσω ενός ερωτήματος SPARQL, το οποίο εκτελεί ο πράκτορας λογισμικού της παραμέτρου.

## 4.2 Ο Προσομοιωτής ASSIST

Προκειμένου να ενισχυθεί η προτεινόμενη προσέγγιση, κατά την οποία ένας πράκτορας λογισμικού (*Device*, *Task* ή *Human Agent*) δημιουργεί κοινωνικούς δεσμούς με άλλους πράκτορας λογισμικού, δημιουργήθηκε το εργαλείο προσομοίωσης *SlOT* [19]. Οι χρήστες αυτού του εργαλείου έχουν τη δυνατότητα να δημιουργήσουν έναν εικονικό κόσμο *SlOT* και να εξετάσουν τη συμπεριφορά των *SlOT Agents* εφαρμόζοντας σημασιολογικούς κανόνες. Για την ανάπτυξη του προσομοιωτή *ASSIST* χρησιμοποιήθηκε το πλαίσιο λογισμικού *Processing* [136] και η υλοποίηση των παραγόντων βασίσθηκε στο πλαίσιο λογισμικού *JADE* [131]. Όσον αφορά την υποστήριξη για το σημασιολογικό ιστό, προτιμήθηκε το πλαίσιο λογισμικού *Jena Apache* [62]. Τέλος, για τη δημοσίευση των θεμάτων και την εγγραφή τους, χρησιμοποιήθηκε το πρωτόκολλο σύνδεσης *MQTT* [46], ενώ το *RabbitMQ* [137] χρησιμοποιήθηκε ως Μεσίτης. Συγκεκριμένα επιλέχθηκε το *MQTT*, αντί άλλων πρωτοκόλλων για το *IoT*, διότι είναι ένα εξαιρετικά ελαφρύ πρωτόκολλο μεταφοράς μηνυμάτων που κάνει χρήση του μοτίβου ανταλλαγής μηνυμάτων *Pub-Sub*.

Το ορατό τμήμα του *ASSIST* είναι το Γραφικό Περιβάλλον Χρήστη (*Graphical User Interface*; *GUI*) του. Το κύριο παράθυρο αποτελείται από δύο περιοχές, την Περιοχή Κοινωνικού Δικτύου (*ΠΚΔ*), όπου εμφανίζονται οι δημιουργούμενες οντότητες (κόμβοι) και οι συνδέσεις μεταξύ τους (ακμές), καθώς και η Περιοχή Διαχείρισης (*ΠΔ*), όπου ο χρήστης μπορεί να δει και να χειρισθεί τις οντότητες και τις συνδέσεις τους.

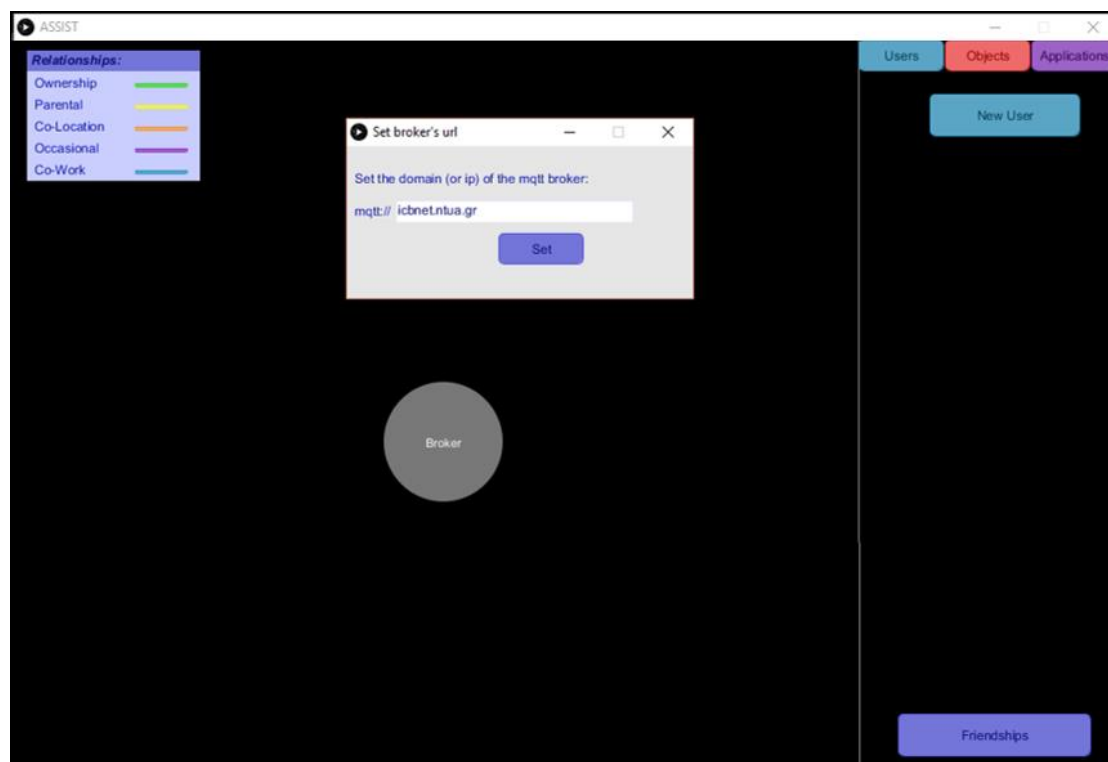
Η *ΠΚΔ* αποτελείται από κόμβους και ακμές. Κάθε κόμβος εμφανίζεται ως κύκλος, ενώ αντιπροσωπεύει μία έξυπνη οντότητα του *IoT*. Οι κόμβοι των χρηστών είναι λευκοί, οι κόμβοι των συσκευών είναι γκρι και οι κόμβοι των εφαρμογών είναι μοβ κύκλοι. Οι ακμές μεταξύ τους είναι μη κατευθυνόμενες γραμμές χρωματισμένες σε σχέση με τον τύπο της σχέσης που αντιπροσωπεύουν. Ένα μικρό πλαίσιο με τα χρώματα και τις ετικέτες των σχέσεων είναι διαθέσιμο στα αριστερά του *ΠΚΔ*, εξηγώντας τους δεσμούς μεταξύ των δημιουργημένων οντοτήτων. Ο κεντρικός γκρι κόμβος αντιπροσωπεύει τον *Broker Agent* (Πράκτορα Μεσίτη).



Εικόνα 45: Το Κοινωνικό Δίκτυο Πρακτόρων Λογισμικού [19].

Ο Broker Agent είναι μία διεπαφή μεταξύ του SIoT Agent και του ASSIST (Εικόνα 45). Οι πληροφορίες που σχετίζονται με τις σχέσεις των αντιπροσώπων αποστέλλονται στον Broker Agent και διαβιβάζονται στο ASSIST. Για παράδειγμα, εάν δύο πράκτορες λογισμικού SIoT γίνουν φίλοι, στέλνουν ένα μήνυμα FIPA στον Broker Agent, ο οποίος στη συνέχεια καλεί μία λειτουργία του ASSIST για να σχεδιάσει μία σύνδεση μεταξύ τους.

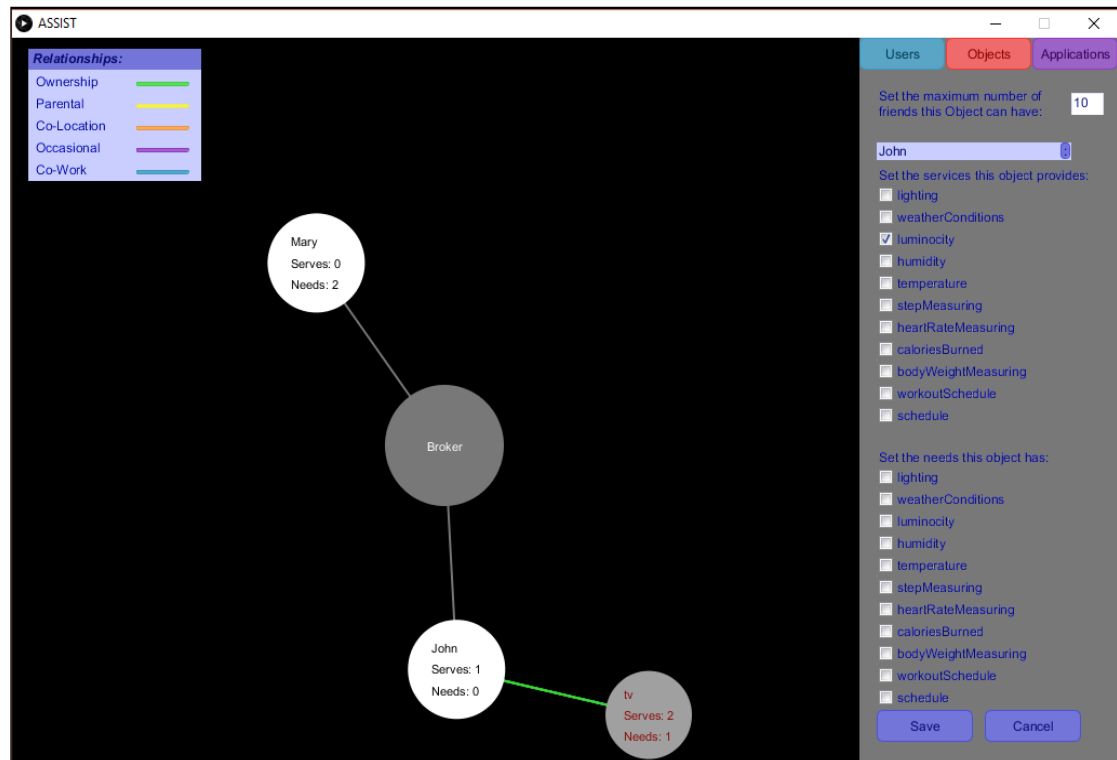
Η ΠΔ αποτελείται από τρεις καρτέλες: α) Χρήστες, β) Αντικείμενα και γ) Εφαρμογές, όπου μπορεί ο χρήστης να προσθέσει μία νέα οντότητα του συγκεκριμένου τύπου. Επιπλέον, στην ίδια περιοχή, παρέχεται ένα κουμπί για τη διαχείριση φιλίας μεταξύ των υφιστάμενων οντοτήτων. Μετά τη δημιουργία του Broker Agent και την προετοιμασία του κύριου παραθύρου, ο χρήστης καλείται να καθορίσει το όνομα τομέα ή τη διεύθυνση IP του MQTT Broker που θα χρησιμοποιηθεί (Εικόνα 46). Μετά την εισαγωγή της διεύθυνσης του MQTT Broker δημιουργείται μία σύνδεση MQTT και εμφανίζεται ο Πράκτορας Μεσίτη, επιτρέποντας στο χρήστη να ξεκινήσει να προσθέτει οντότητες στο κύριο παράθυρο. Η πρώτη οντότητα, ακολουθώντας αυτή του Broker Agent, πρέπει να είναι μία οντότητα χρήστη, επομένως όλες οι εφαρμογές και τα αντικείμενα που κατέχει θα κληρονομήσουν τη διεύθυνση IP του Πράκτορα Μεσίτη. Ένα όνομα, ένας μέγιστος αριθμός φίλων και δύο λίστες (δηλ., μία από τις διαθέσιμες υπηρεσίες και μία από τις διαθέσιμες ανάγκες της οντότητας) είναι τα δεδομένα που πρέπει να υποβληθούν πριν από τη δημιουργία και τη σύνδεση μεταξύ του νέου χρήστη και του Broker Agent. Μετά τη δημιουργία οποιασδήποτε οντότητας, αυτή εκδίδει τις υπηρεσίες της στο MQTT Broker κάνοντας χρήση του πρωτοκόλλου MQTT.



Εικόνα 46: Screenshot που δείχνει την καρτέλα που απαιτεί την εισαγωγή της διεύθυνσης URL του MQTT Broker στο ASSIST.

Έχοντας τουλάχιστον ένα διαθέσιμο User Agent, ο χρήστης μπορεί να προσθέσει ένα νέο χρήστη, αντικείμενο ή εφαρμογή. Στην περίπτωση των δύο τελευταίων, στο διάλογο για το όνομα της οντότητας υπάρχει, επίσης, η δυνατότητα παροχής του κωδικού EPC της οντότητας. Επιπλέον, παρέχεται το πεδίο του μέγιστου αριθμού των φίλων που μπορεί να έχει η οντότητα και οι κατάλογοι των υπηρεσιών και των αναγκών, ενώ, όπως φαίνεται στην Εικόνα 47, πρέπει να επιλεγθεί και ο ιδιοκτήτης της οντότητας μεταξύ των υπάρχοντων χρηστών.

Αν και οι φιλίες των οντοτήτων διεκπεραιώνονται από τους αντίστοιχους πράκτορες λογισμικού αυτόματα, μπορεί κανείς να προσθέσει μία νέα φιλία μεταξύ δύο οντοτήτων, η οποία μπορεί να είναι είτε μία Περιστασιακή Σχέση Αντικειμένου είτε μία Σχέση Συνεγκατάστασης Αντικειμένου, καθορίζοντας τα δύο μέρη της φιλίας (π.χ., οποιοσδήποτε από τους υπάρχοντες χρήστες, αντικείμενα ή εφαρμογές) μαζί με τον τύπο αυτής της νέας φιλίας. Μετά τη δημιουργία της νέας φιλίας, προστίθεται μία νέα σύνδεση μεταξύ των δύο κόμβων. Το χρώμα του συνδέσμου εξαρτάται από τον τύπο της σχέσης, ενώ περισσότερες λεπτομέρειες για μία οντότητα παρέχονται κάνοντας κλικ σε αυτόν. Όλες οι πληροφορίες μίας οντότητας, συμπεριλαμβανομένου του ονόματος, των υπηρεσιών και των αναγκών της μαζί με τη λίστα των φίλων της, εμφανίζονται στην ΠΔ.



Εικόνα 47: Το GUI του ASSIST κατά τη δημιουργία ενός νέου αντικειμένου.



## 5 Αναγνώριση Ανθρώπινης Δραστηριότητας με χρήση του μοντέλου PerceptionNet

Στο παρόν κεφάλαιο εισάγεται η ιδέα της εφαρμογής καθυστερημένης συγχώνευσης στα δεδομένα κίνησης, μέσω μίας 2D συνέλιξης, σε μία προσπάθεια να ανακαλυφθούν πιο σύνθετα πρότυπα δραστηριότητας, που προέρχονται από τη διασταυρούμενη συσχέτιση μεταξύ των αισθητήρων κίνησης. Επιλέχθηκε ένα βαθύ ConvNet, το οποίο ονομάζεται PerceptionNet, για την αυτόματη εξαγωγή χαρακτηριστικών και την ταξινόμησή τους, επειδή το συγκεκριμένο μοντέλο εκμεταλλεύεται τη σημασιολογία και την τοπολογία των δεδομένων εισόδου σε σύγκριση με τα πλήρως συνδεδεμένα Νευρωνικά Δίκτυα. Τέλος, για να προστεθεί ένα επίπεδο 2D συνέλιξης, το μοντέλο PerceptionNet δέχεται ως είσοδο σήματα κίνησης στοιβαγμένα κατακόρυφα (vertically stacked).

### 5.1 Ρύθμιση Πειραμάτων

Τα πειράματα εκτελέστηκαν σε σταθμό εργασίας εξοπλισμένο με NVIDIA GTX Titan X GPU, ο οποίος έχει RAM 12 gigabytes, 3072 πυρήνες CUDA και εύρος ζώνης 336,5 GB/s. Χρησιμοποιήθηκε η Python ως γλώσσα προγραμματισμού, και η βιβλιοθήκη NumPy για πολλαπλασιασμούς πινάκων, προεπεξεργασία δεδομένων και κατακερματισμό δεδομένων. Για τη σχεδίαση και εκπαίδευση του προτεινόμενου μοντέλου PerceptionNet χρησιμοποιήθηκε η βιβλιοθήκη νευρωνικών δικτύων υψηλού επιπέδου Keras, έχοντας ως backend τη βιβλιοθήκη Theano. Προκειμένου να επιταχυνθούν οι πολλαπλασιασμοί των τανυστών, χρησιμοποιήθηκε το CUDA Toolkit σε συνδυασμό με τη cuDNN, η οποία είναι η βιβλιοθήκη της NVIDIA για την επιτάχυνση των συναρτήσεων των νευρωνικών δικτύων στην κάρτα γραφικών. Το παραπάνω λογισμικό εγκαταστάθηκε σε ένα λειτουργικό σύστημα Ubuntu Linux 16.04.

#### 5.1.1 Σύνολα Δεδομένων

Το μοντέλο PerceptionNet αξιολογήθηκε με βάση δύο δημόσια διαθέσιμα σύνολα δεδομένων AAΔ, το UCL [138] και το PAMAP2 [139]. Το πρώτο, χρησιμοποιήθηκε για να συντονισθούν οι υπερ-παραμέτροι (Πίνακας 6), και στη συνέχεια συγκρίθηκαν οι επιδόσεις του PerceptionNet με τις τεχνολογίες αιχμής: α) CNN [118], β) LSTM [126] γ) CNN σε φασματογράφημα [123], και δ) μέθοδο SVM βασισμένη σε χαρακτηριστικά εξαχθέντα από ανθρώπους [138]. Τέλος, προκειμένου να εξετασθεί η επεκτασιμότητα της παρούσας προσέγγισης, χρησιμοποιήθηκε η ίδια αρχιτεκτονική (PerceptionNet) στο σύνολο δεδομένων PAMAP2 και συγκρίθηκε με τις συνήθεις μεθόδους CNN [118] και Convolutional LSTM [126].

### 5.1.1.1 UCL

Το σύνολο δεδομένων UCL για ΑΑΔ αποτελείται από δεδομένα αισθητήρα τριαξονικού επιταχυνσιόμετρου και γυροσκοπίου, τα οποία συλλέγονται από ένα έξυπνο κινητό τηλέφωνο (smartphone Samsung Galaxy S II) που είναι τοποθετημένο στη μέση των χρηστών. Μία ομάδα 30 εθελοντών, ηλικίας 19 έως 48 ετών, εκτέλεσε έξι καθημερινές δραστηριότητες (*στέκομαι, κάθομαι, είμαι ξαπλωμένος, περπατάω, ανεβαίνω σκάλες και κατεβαίνω σκάλες*). Οι κινητοί αισθητήρες παρήγαγαν δεδομένα 3-αξονικής γραμμικής επιτάχυνσης και 3-αξονικής γωνιακής ταχύτητας με ρυθμό δειγματοληψίας 50 Hz και τμηματοποιήθηκαν σε χρονικά παράθυρα 128 τιμών (2,56 sec), με επικαλύψεις 50%.

Το λαμβανόμενο σύνολο δεδομένων περιέχει 10.299 δείγματα, τα οποία χωρίζονται σε δύο σειρές, όπου το 70% των δεδομένων των εθελοντών (21 εθελοντές) επιλέχθηκαν για τη δημιουργία των δεδομένων εκπαίδευσης (7.352 δείγματα) και το 30% (9 εθελοντές) για τη δημιουργία των δεδομένων δοκιμής (2.947 δείγματα). Επιπλέον, ακολουθώντας τα αποτελέσματα της [140], όπου αποδείχθηκε ότι θα πρέπει να εφαρμόζονται τεχνικές επικύρωσης (validation) που δεν είναι εξαρτώμενες από τους χρήστες για την αξιολόγηση των συστημάτων παρακολούθησης της δραστηριότητας για το συντονισμό των υπερ-παραμέτρων, στη διάρκεια της φάσης επικύρωσης τα δεδομένα προερχόμενα από τρεις χρήστες χρησιμοποιήθηκαν ως δεδομένα επικύρωσης (*Leave-3-Subject-Out*). Συγκεκριμένα, τα δείγματα 3 εθελοντών (27, 29 και 30) χρησιμοποιήθηκαν ως σύνολο επικύρωσης, ο αριθμός των οποίων ισούται με το 15% του σετ εκπαίδευσης.

Τέλος, κανονικοποιήθηκαν οι τιμές του κάθε σήματος ξεχωριστά, αφαιρώντας από αυτές τη συνολική μέση τιμή και διαιρώντας το αποτέλεσμα με την τυπική απόκλιση:

$$z_x = \frac{x - \mu_x}{\sigma_x} \quad (4.1)$$

### 5.1.1.2 PAMAP2

Το σύνολο δεδομένων PAMAP2 για ΑΑΔ περιλαμβάνει 12 δραστηριότητες σχετικές με τον τρόπο ζωής των χρηστών (όπως περπάτημα, ποδηλασία, σιδέρωμα, κ.λπ.) εκτελεσθείσες από 9 συμμετέχοντες που φορούν 3 ασύρματες, αδρανειακές μονάδες μέτρησης (Inertial Measurement Units; IMU) της εταιρείας Colibri και έναν αισθητήρα παρακολούθησης καρδιακών παλμών. Οι 3 μονάδες IMU είχαν συχνότητα δειγματοληψίας 100 Hz, τοποθετήθηκαν στον κυρίαρχο βραχίονα, στο θώρακα και στον αστράγαλο της κυρίαρχης πλευράς και παρήγαγαν δεδομένα τριαξονικού επιταχυνσιόμετρου, γυροσκοπίου και μαγνητόμετρου. Προκειμένου να επιτευχθεί ο ίδιος ρυθμός δειγματοληψίας με το σύνολο δεδομένων UCL και ο ίδιος τύπος



σήματος αισθητήρων, το σύνολο δεδομένων PAMAP2 υποδειγματίστηκε στα 50Hz και επιλέχθηκαν μόνο τα δεδομένα του επιταχυνσιόμετρου και του γυροσκοπίου. Το σύνολο των δεδομένων που προέκυψε είχε 18 διαστάσεις, με το ίδιο χρονικό παράθυρο (2,56 δευτερόλεπτα) και επικάλυψη (50%) με το σύνολο δεδομένων UCL.

Δεδομένου ότι τα δεδομένα συλλέχθηκαν μόνο από 9 συμμετέχοντες και επειδή μόνο 4 άτομα (1, 2, 5 και 8) είχαν αρκετά δείγματα από όλες τις δραστηριότητες, επιλέχθηκε μία προσέγγιση *Leave-1-Subject-Out*, για τα σύνολα δεδομένων δοκιμής και επικύρωσης. Πιο συγκεκριμένα, τα δείγματα του συμμετέχοντος 1 χρησιμοποιήθηκαν για το σετ δοκιμών και τα δείγματα του συμμετέχοντος 5 για το σύνολο επικύρωσης. Το σετ εκπαίδευσης περιέχει 13.980, το σετ δοκιμών 2.453 και το σετ επικύρωσης 2.688 δείγματα. Τα δείγματα PAMAP2, επίσης, κανονικοποιήθηκαν χρησιμοποιώντας την (2.13).

### 5.1.2 Μετρικές Απόδοσης

Ως μετρικές απόδοσης χρησιμοποιήθηκαν τα εξής: ακρίβεια (precision), ανάκληση (recall), σταθμισμένη βαθμολογία F1 (weighted F1-score) και ορθότητα (accuracy). Πρέπει να σημειωθεί ότι η ορθότητα, σε αντίθεση με τις άλλες 3 μετρήσεις, λαμβάνει υπόψη μόνο το συνολικό αριθμό των δειγμάτων και όχι την ταξική ανισορροπία (class imbalance). Από την άλλη πλευρά, η ακρίβεια, η ανάκληση και η βαθμολογία F1 θεωρούν το συνολικό αριθμό δειγμάτων για κάθε κατηγορία χωριστά. Οι παραπάνω μετρήσεις περιγράφονται ως:

$$accuracy = \frac{1}{N} \sum_{i=1}^N (TP_i + TN_i) \quad (4.2)$$

$$precision = \frac{1}{N} \sum_{i=1}^N \left( \frac{TP_i}{TP_i + FP_i} \right) \quad (4.3)$$

$$recall = \frac{1}{N} \sum_{i=1}^N \left( \frac{TP_i}{TP_i + FN_i} \right) \quad (4.4)$$

$$F_1 = \sum_i 2 * w_i \frac{precision_i * recall_i}{precision_i + recall_i} \quad (4.5)$$

όπου οι όροι  $TP, TN, FP, FN$  αντιπροσωπεύουν τις αληθώς θετικές (True Positive), τις αληθώς αρνητικές (True Negative), τις ψευδώς θετικές (False Positive) και τις ψευδώς αρνητικές (False Negative) προβλέψεις αντιστοίχως. Ο δείκτης  $i$  αντιστοιχεί στο δείκτη της κλάσης και ο όρος  $w_i = n_i/N$  είναι το ποσοστό των δειγμάτων της

κλάσης  $i$ , με το  $n_i$  να αντιπροσωπεύει τον αριθμό των δειγμάτων της κλάσης  $i$  και το  $N$  τον αριθμό των συνολικών δειγμάτων.

Επιπλέον, επιλέχθηκε ο πίνακας σύγχυσης (confusion matrix) ως απεικόνιση της απόδοσης ταξινόμησης του PerceptionNet. Ο πίνακας σύγχυσης είναι εύκολο να ερμηνευθεί: δείχνει πού ο αλγόριθμος ταξινόμησης "συγχέει" μία κλάση με μία άλλη (π.χ., προβλέπει τη δραστηριότητα *κάθομαι* αντί της αληθινής που είναι *στέκομαι*). Από μαθηματική σκοπιά, ο πίνακας σύγχυσης μπορεί να περιγραφεί από το  $M_{ij}$ , με το  $i$  να δηλώνει τις "πραγματικές" κατηγορίες και το  $j$  τις "προβλεπόμενες". Αθροίζοντας όλες τις καταχωρίσεις στη σειρά  $i$  του πίνακα, εμφανίζεται ο συνολικός αριθμός των δειγμάτων που σημειώνονται ως δραστηριότητα  $i$ , ενώ με άθροιση όλων των καταχωρίσεων στη στήλη  $j$  του πίνακα εμφανίζεται ο συνολικός αριθμός των δειγμάτων που προβλέπονται ως δραστηριότητα  $j$ .

**Πίνακας 6: Πειραματική ρύθμιση των υπερ-παραμέτρων του συνόλου δεδομένων UCL [141]**

Σύμβολο	Υπερ-παραμέτρος	Τιμή
-	batch size	64
$\alpha$	learning rate	1.0
$\rho$	Rho	0.95
$\epsilon$	Epsilon	1e-08
-	number of channels	1
-	input height	6
-	input width	128
-	number of convolutional layers	3
-	1D convolution size	1X5-1X17
-	2D convolution size	3X15
-	2D strides size	3X1
-	1D max pooling size	1X2
-	1D max pooling stride	1X2
-	Dropout	0-0.7
-	activation map channels	32-192
-	dense layer size	0-1500
-	2D convolutional layer	1-3
-	maximum epochs	2000
-	early stopping criterion epochs	100

## 5.2 Αλγόριθμος Βελτιστοποίησης

Ο αλγόριθμος Adadelta [142] επιλέχθηκε ως βελτιστοποιητής του δικτύου. Ο Adadelta προσαρμόζεται δυναμικά στις επαναλήψεις, χωρίς να χρειάζεται χειροκίνητο συντονισμό του ρυθμού εκμάθησης και εμφανίζει ανθεκτικότητα σε: α)

θορυβώδεις πληροφορίες των παραγώγων, β) διαφορετικές επιλογές αρχιτεκτονικής μοντέλου, γ) διάφορες μορφές δεδομένων, και δ) επιλογή υπερ-παραμέτρων (hyper-parameters). Σύμφωνα με την [142], έστω ότι  $L'(\theta_t)$  είναι η πρώτη παράγωγος της συνάρτησης κόστους  $f$  σε συνάρτηση με τις παραμέτρους  $\theta$  στο χρονικό βήμα  $t$ , τότε ο όρος  $g_t$  αποτελεί το δεύτερο στιγμιότυπο της. Με δεδομένο τον όρο αποσύνθεσης  $\rho$  και την αντιστάθμιση  $\varepsilon$  πραγματοποιούνται οι ακόλουθες ενημερώσεις:

$$g_t = (1 - \rho)L'(\theta_t)^2 + \rho g_{t-1} \quad (4.6)$$

όπου  $g_0 = 0$  και  $s_0 = 0$ . Ο όρος  $s_t$  υποδηλώνει τη 2η στιγμή του  $\Delta\theta_t^2$  για την ενημέρωση των παραμέτρων:

$$\Delta\theta_t = - \frac{\sqrt{s_{t-1} + \varepsilon}}{\sqrt{g_t + \varepsilon}} L'(\theta_t) \quad (4.7)$$

$$s_t = (1 - \rho)\Delta\theta_t^2 + \rho s_{t-1} \quad (4.8)$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t \quad (4.9)$$

## 5.3 Αποτελέσματα

### 5.3.1 Φάση Επικύρωσης

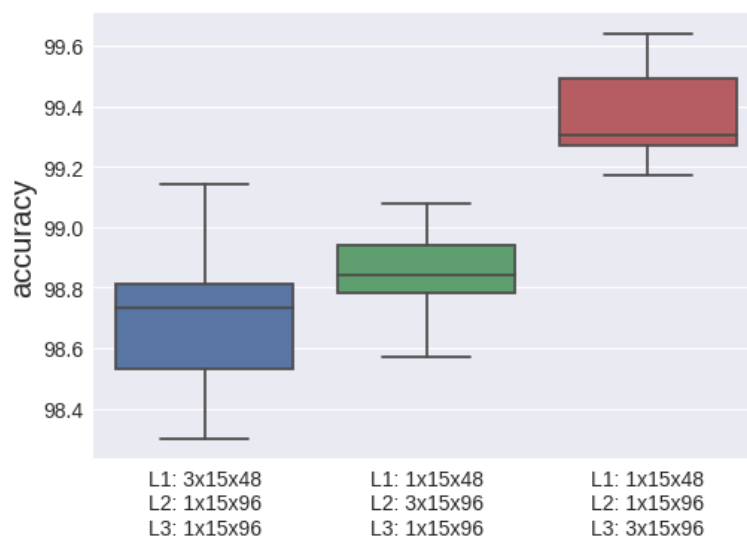
Πριν από τη δοκιμασία της παρούσας προσέγγισης, χρησιμοποιήθηκε το σύνολο επικύρωσης του UCL για να συντονισθούν οι υπερ-παραμέτροί του. Ο Πίνακας 6 περιέχει όλες τις υπερ-παραμέτρους και τις πιθανές τιμές τους. Δεδομένου ότι είναι εξαιρετικά χρονοβόρο να εκπαιδευθεί ένα μοντέλο βαθύς CNN και επειδή το σύνολο δεδομένων UCL εξετάσθηκε διεξοδικά στην [118], δεν αξιολογήθηκαν εξαντλητικά όλοι οι πιθανοί συνδυασμοί των υπερ-παραμέτρων. Έτσι, βασίσθηκε η επιλογή του πιο ελπιδοφόρου μοντέλου στους πιο σημαντικούς παράγοντες (μέγεθος φίλτρου, πιθανότητα dropout, στρώμα που θα εκτελεσθεί η δισδιάστατη συνέλιξη και χρήση πλήρως συνδεδεμένου στρώματος ή ολικής υποδειγματοληψίας μέσης τιμής) που διαφοροποιούν την παρούσα μέθοδο από αυτή που περιγράφεται στην [118].

Δεδομένου ότι η σύγκλιση και η ορθότητα της πρόβλεψης ενός βαθύς νευρωνικού δικτύου εξαρτώνται πολύ από την αρχικοποίηση των βαρών [143], για να επιτευχθούν πιο αντιπροσωπευτικά αποτελέσματα για κάθε υπερ-παραμέτρο, εκτελέσθηκαν τα πειράματα 10 φορές. Η μέση τιμή κάθε διαφορετικού υπερ-παραμετρικού μοντέλου χρησιμοποιήθηκε ως κριτήριο επιλογής. Θα πρέπει να σημειωθεί ότι επιλέχθηκε μία παραλλαγή της αρχικοποίησης βάρους που εισήχθη από τον K. He και λοιπούς [144] (δηλ., οι τυχαίοι αριθμοί ελήφθησαν από

ομοιόμορφη κατανομή αντί από μία κανονική κατανομή), των οποίων τα ανώτερα και κατώτερα όρια δίδονται από τη συνάρτηση:

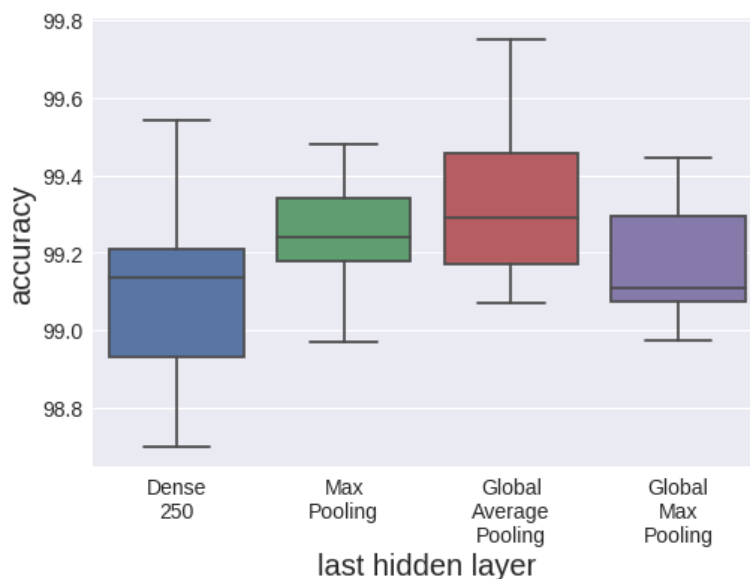
$$w_i = \pm \sqrt{\frac{2}{N_{in}}} \quad (4.10)$$

όπου το  $N_{in}$  αντιπροσωπεύει το συνολικό αριθμό των νευρώνων που εξάγει το ( $i - 1$ ) στρώμα, που είναι η είσοδος για το επόμενο στρώμα. Επιπλέον, ο ελάχιστος αριθμός εποχών (δηλ., πόσες φορές θα χρησιμοποιηθούν όλα τα δεδομένα εκπαίδευσης) ορίστηκε στις 2.000, αλλά η διαδικασία εκπαίδευσης τερματιζόταν αυτόματα αν η μέγιστη τιμή της ορθότητας εκπαίδευσης δεν είχε βελτιωθεί μετά από 100 εποχές. Το μοντέλο που πέτυχε το χαμηλότερο ποσοστό σφάλματος στο σύνολο των δειγμάτων επικύρωσης καταχωρήθηκε σε ένα αρχείο και τα φίλτρα του χρησιμοποιήθηκαν για να αποκτήσουν την ορθότητα του μοντέλου στο σετ των δειγμάτων δοκιμής.

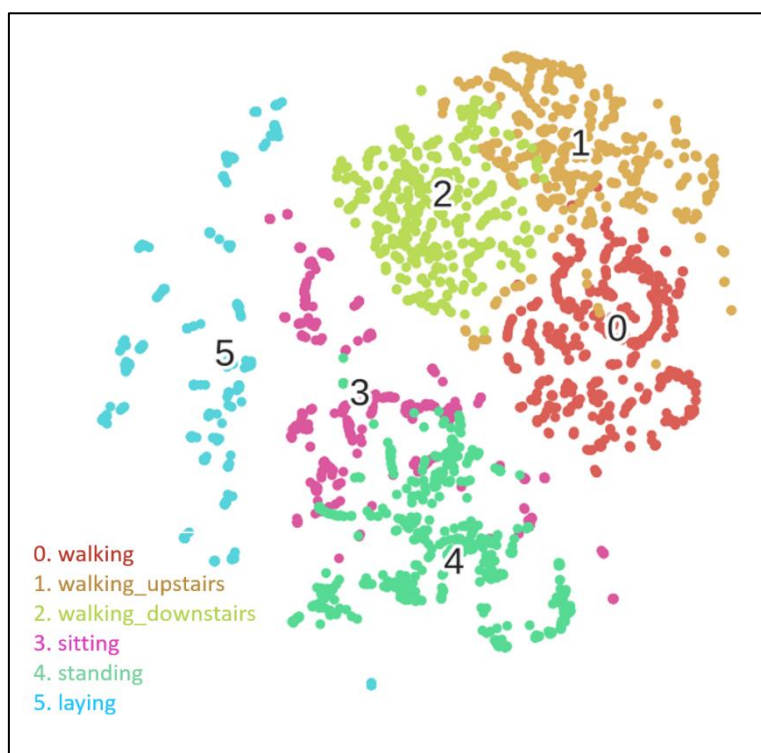


Εικόνα 48: Τα αποτελέσματα ορθότητας της χρήσης δισδιάστατης συνέλιξης στο 1<sup>ο</sup>, 2<sup>ο</sup> και 3<sup>ο</sup> επίπεδο συνέλιξης του UCL συνόλου επικύρωσης [141].

Το βέλτιστο μοντέλο που ανακτήθηκε αποτελείται από τρία συνελκτικά επίπεδα (L1: 1x15x48, L2: 1x15x96 και L3: 3x15x96), το καθένα ακολουθούμενο από μία συνάρτηση ενεργοποίησης Relu, μέγεθος υποδειγματοληψίας μέγιστης τιμής (1x2) και μία υψηλή πιθανότητα απόρριψης νευρώνων, ίση με 0,4. Πρέπει να σημειωθεί ότι εφαρμόστηκαν οι δισδιάστατες συνέλιξεις σε διαφορετικά στρώματα, καθένα από τα οποία είχε μέγεθος 3x15 και βήμα (stride) ίσο με 3 και 1, για τον κάθετο και τον οριζόντιο άξονα αντίστοιχα. Η Εικόνα 48 δείχνει ότι η εφαρμογή της 2D συνέλιξης στο τελευταίο συνελκτικό στρώμα είναι πολύ πιο αποτελεσματική.



Εικόνα 49: Τα αποτελέσματα ορθότητας βασισμένα στη χρήση διαφορετικών τελευταίων κρυφών επιπέδων του UCL συνόλου επικύρωσης.



Εικόνα 50: Η t-SNE οπτικοποίηση του τελευταίου κρυφού επιπέδου στο μοντέλο PerceptionNet για τις έξι δραστηριότητες [141].

Επιπλέον, εξετάστηκε η χρήση του πλήρως συνδεδεμένου στρώματος νευρώνων, ως τελευταίου κρυφού επιπέδου, αλλά και η προσθήκη μίας συνάρτησης ολικής υποδειγματοληψίας μέσης τιμής ή μίας ολικής υποδειγματοληψίας μέγιστης τιμής. Όπως φαίνεται στην Εικόνα 49, η υψηλότερη μέση ορθότητα ανακτήθηκε

κάνοντας χρήση συνάρτησης ολικής υποδειγματοληψίας μέσης τιμής (99,38%). Η Εικόνα 50 δείχνει ότι εφαρμόζοντας τον αλγόριθμο δισδιάστατης οπτικοποίησης t-SNE [145] μετά το τελευταίο στρώμα συνέλιξης (Παράρτημα A.2 Γραφικές παραστάσεις των σημάτων κίνησης κατά τη διάρκεια επεξεργασίας τους) τα περισσότερα από τα δείγματα των έξι κατηγοριών δραστηριότητας είναι εμφανώς διαχωρισμένα.

### 5.3.2 Αρχιτεκτονική Βαθούς Συνελικτικού Νευρωνικού Δικτύου

Η προτεινόμενη αρχιτεκτονική Βαθούς Συνελικτικού Νευρωνικού Δικτύου αποτελείται από τα παρακάτω στρώματα:

*1<sup>ο</sup> Στρώμα:* 48 συνελικτικά φίλτρα με μέγεθος (1, 15), δηλ., το φίλτρο  $W^1$  έχει μέγεθος (1, 15, 1, 48), ενώ η αρχικοποίησή του έγινε κάνοντας χρήση της εξίσωσης (4.6). Μετά τη μονοδιάστατη συνέλιξη ακολουθεί η συνάρτηση ενεργοποίησης ReLu, και εφαρμόζεται μονοδιάστατη υποδειγματοληψία μέγιστης τιμής (1, 2), με βήμα (1, 2). Τέλος, εφαρμόζεται απόρριψη συνδέσεων dropout, με πιθανότητα απόρριψης ίση με 0.40.

*2<sup>ο</sup> Στρώμα:* 96 συνελικτικά φίλτρα με μέγεθος (1, 15), δηλ., το φίλτρο  $W^2$  έχει μέγεθος (1, 15, 48, 96), ενώ η αρχικοποίησή του έγινε κάνοντας χρήση της εξίσωσης (4.6). Μετά τη μονοδιάστατη συνέλιξη ακολουθεί η συνάρτηση ενεργοποίησης ReLu, και εφαρμόζεται μονοδιάστατη υποδειγματοληψία μέγιστης τιμής (1, 2), με βήμα (1, 2). Τέλος, εφαρμόζεται απόρριψη συνδέσεων dropout, με πιθανότητα απόρριψης ίση με 0.40.

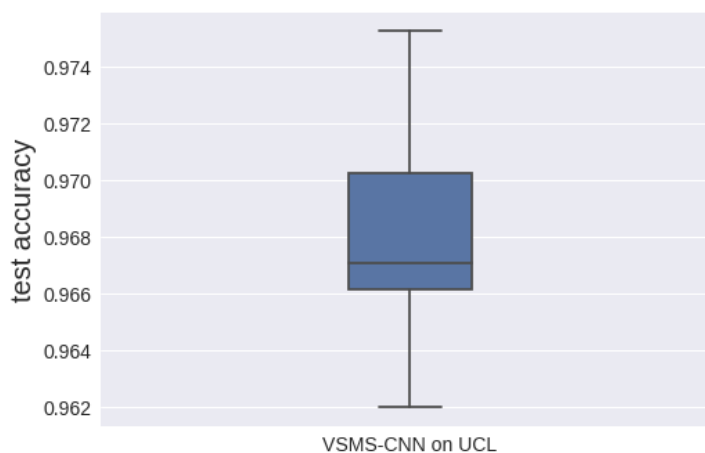
*3<sup>ο</sup> Στρώμα:* 96 συνελικτικά φίλτρα με μέγεθος (3, 15), δηλ., το φίλτρο  $W^3$  έχει μέγεθος (3, 15, 96, 96), ενώ η αρχικοποίησή του έγινε κάνοντας χρήση της εξίσωσης (4.10). Μετά τη δισδιάστατη συνέλιξη ακολουθεί η συνάρτηση ενεργοποίησης ReLu, και εφαρμόζεται ολική υποδειγματοληψία μέγιστης τιμής. Τέλος, εφαρμόζεται απόρριψη συνδέσεων dropout, με πιθανότητα απόρριψης ίση με 0.40.

*4<sup>ο</sup> Στρώμα:* 6 πλήρως συνδεδεμένοι νευρώνες (στρώμα εξόδου), δηλ., το βάρος  $W^4$  έχει μέγεθος (96, 6), ενώ ακολουθεί μία συνάρτηση ενεργοποίησης Softmax (εξίσωση 2.6).

Τέλος, για τη βελτιστοποίηση του μοντέλου, εφαρμόστηκε αλγόριθμος Adadelta, με βαθμό εκμάθησης ίσο με 1, όρο αποσύνθεσης  $\rho$  ίσο με 0.95 και αντιστάθμιση  $\epsilon$  ίση με  $1e-08$ .

### 5.3.3 Φάση δοκιμής για το σύνολο δεδομένων UCL

Τα αποτελέσματα που ελήφθησαν χρησιμοποιώντας το μοντέλο PerceptionNet στο σετ δοκιμών παρουσιάζονται στην Εικόνα 51 και εντοπίζονται στην περιοχή του 0,9620 έως 0,9752. Τέλος, μετά την απόκτηση των αποτελεσμάτων, αναπτύχθηκε ένα σύνολο από μοντέλα PerceptionNet στα πλαίσια της χρήσης Συλλογικής Μάθησης (Ensemble Learning), βασισμένο στην ψηφοφορία βάσει των συνολικών πιθανοτήτων. Πιο συγκεκριμένα, προστέθηκε η πιθανότητα των 10 ξεχωριστών μοντέλων και διαιρέθηκε έπειτα με 10 (σύνολο των μοντέλων), επιλέγοντας στη συνέχεια μέγιστη μέση πιθανότητα (probability voting; Εικόνα 52). Η καλύτερη εποχή στα δεδομένα επικύρωσης πέτυχε ακρίβεια 0,9725 στα δεδομένα των δοκιμών (Παράρτημα Α.1 Παρουσίαση κώδικα για Αναγνώριση Ανθρώπινης Δραστηριότητας (PerceptionNet), η οποία είναι περίπου 2,5% υψηλότερη από το ConvNet της [118]. Ο Πίνακας 7 συγκρίνει την ορθότητα που προέκυψε από το δισδιάστατο μοντέλο ConvNet έναντι προηγούμενων βιβλιογραφικών αποτελεσμάτων που αναφέρθηκαν στη βιβλιογραφία και την εφαρμογή του Convolutional LSTM.



Εικόνα 51: Ορθότητα δοκιμής για τα μοντέλα PerceptionNet όσον αφορά το σύνολο δεδομένων UCL [141].

---

```

function probability_ensemble(trained_cnn_models, num_classes, test_data):
    foreach i belonging in test_data
        foreach j belonging in num_classes
            foreach k belonging in trained_cnn_models
                total_preds(i, j, k) ← predict(i, j, k)
    mean_total_preds ← meantrained_cnn_models(total_preds)
    ensemble_preds ← argmaxtrained_cnn_models(mean_total_preds)
    return ensemble_preds
    
```

---

Εικόνα 52: Ψευδοκώδικας πιθανοτικής Συλλογικής Μάθησης.

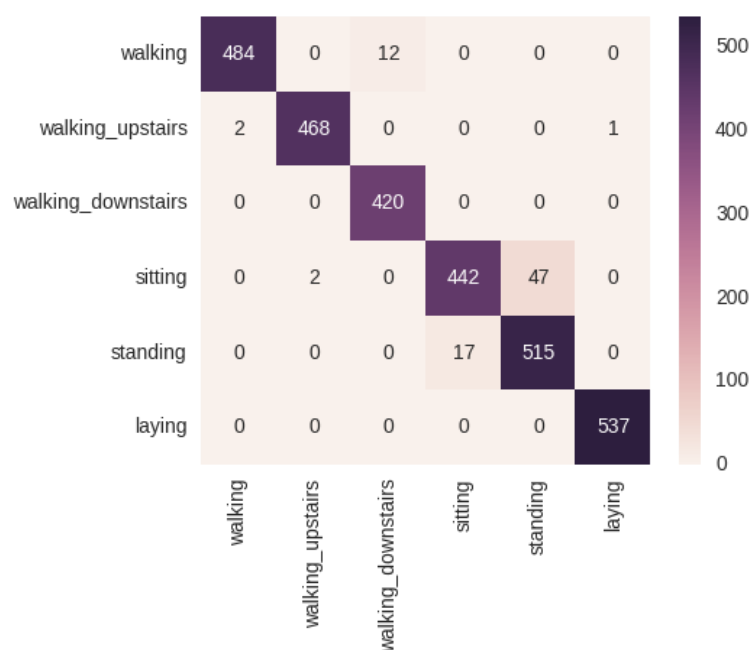
Πίνακας 7: Σύγκριση του μοντέλου PerceptionNet με άλλα μοντέλα τεχνολογιών αιχμής [141].

METHOD	Accuracy on test data
CNN [118]	94.79%

CNN + FFT features [118]	95.75%
SVM on HCF [138]	96.00%
CNN on spectrogram [123]	95.18%
Convolutional LSTM	92.59%
PerceptionNet	<b>97.25%</b>

Πίνακας 8: Ορθότητα δοκιμής ανά συμμετέχοντα του συλλογικού μοντέλου PerceptionNet [141].

Συμμετέχοντας	Accuracy on test data
2	98.01%
4	98.10%
9	90.95%
10	92.49%
12	96.24%
13	100.0%
18	99.45%
20	98.01%
24	100.0%



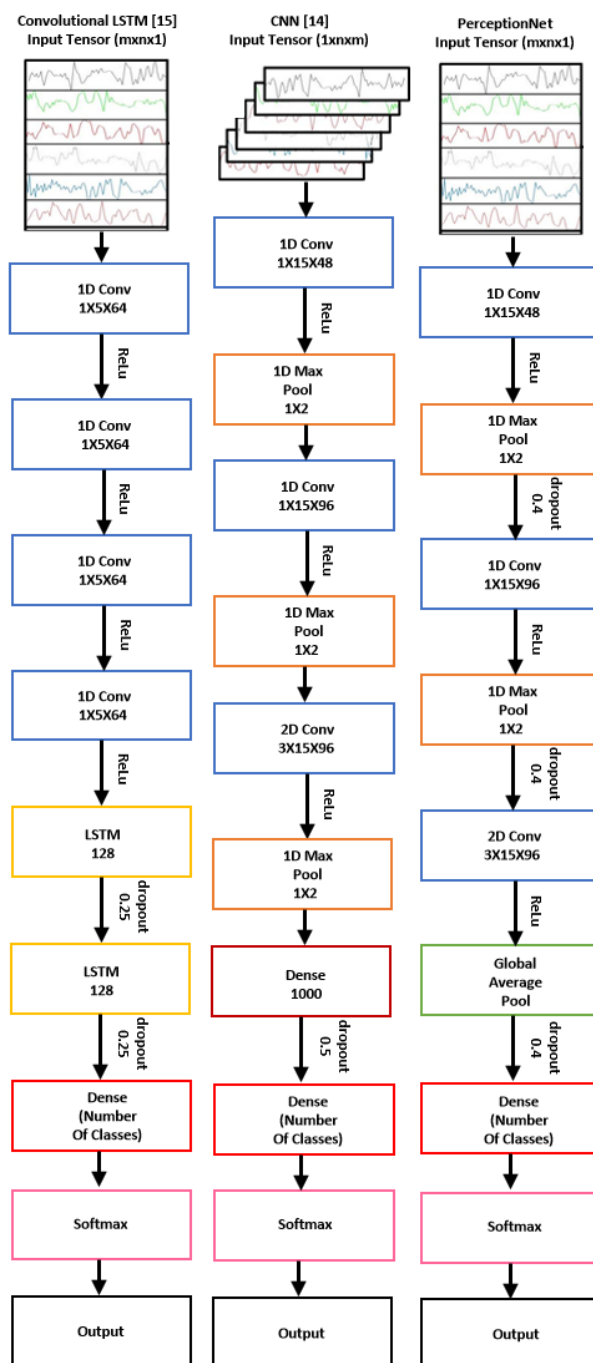
Εικόνα 53: Ο πίνακας σύγχυσης του συλλογικού μοντέλου PerceptionNet για το σύνολο δεδομένων UCL [141].

Η ακρίβεια ανά συμμετέχοντα του συλλογικού 2D ConvNet παρουσιάζεται στον Πίνακας 8, ενώ στην Εικόνα 53 παρουσιάζεται ο πίνακας σύγχυσης (ακρίβεια: 0.9731, ανάκληση: 0.9725, και σταθμισμένη βαθμολογία F1: 0.9724), γεγονός που αποκαλύπτει τη δυσκολία διακρίσεως της δραστηριότητας στέκομαι από τη δραστηριότητα κάθομαι και το αντίστροφο.



### 5.3.4 Φάση δοκιμής για το σύνολο δεδομένων PAMAP2

Στην Εικόνα 54 παρουσιάζονται η αρχιτεκτονική του μοντέλου PerceptionNet και αυτή των άλλων μοντέλων Βαθιάς Μάθησης (δηλαδή το σύνθετες CNN και το Convolutional LSTM που προτείνουν οι έρευνες [118] και [126] αντίστοιχα) που εφαρμόστηκαν για τη σύγκριση του παραπάνω μοντέλου με βάση το σύνολο δεδομένων PAMAP2.



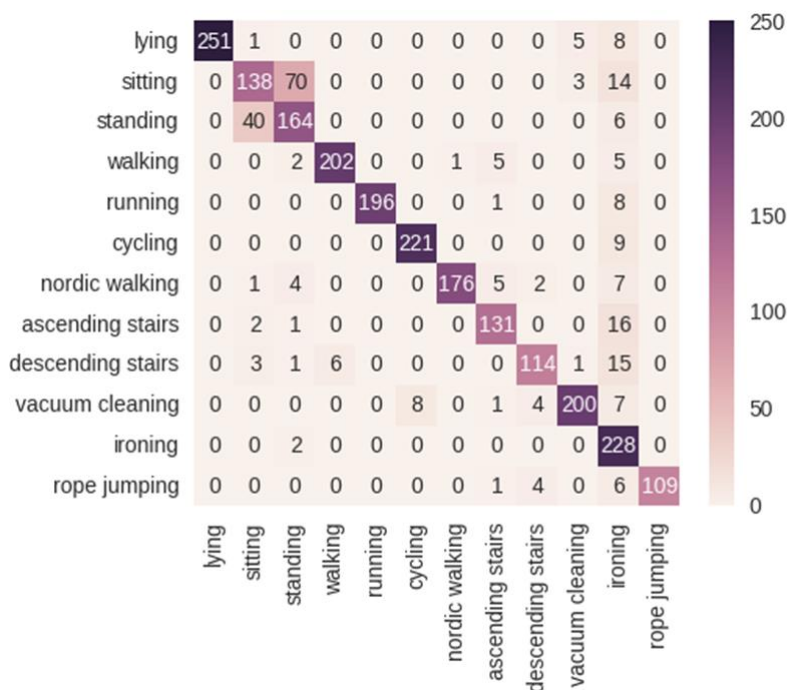
Εικόνα 54: Αρχιτεκτονικές μοντέλων Βαθιάς Μάθησης για αναγνώριση ανθρώπινης δραστηριότητας. Η 1<sup>η</sup> αφορά το μοντέλο Convolutional LSTM που παρουσιάζεται στην [126], η 2<sup>η</sup> τη συνήθη CNN που παρουσιάζεται στην [118] και η 3<sup>η</sup> αφορά την προσέγγιση PerceptionNet [141].

Πρέπει να σημειωθεί ότι η είσοδος είναι ένας τανυστής (πολυδιάστατος πίνακας)  $h \times d \times q$ . Αρχικά υποδηλώνεται το ύψος  $h$ , το μήκος  $d$ , και τέλος το βάθος  $q$  του τανυστή. Στην περίπτωση του PerceptionNet, ο τανυστής εισόδου έχει μέγεθος  $m \times n \times 1$ , όπου το σύμβολο  $m$  ισούται με τον αριθμό των σημάτων και το σύμβολο  $n$  με τον αριθμό των χρονικών καρέ του σήματος. Σύμφωνα με την [118], ο τανυστής εισόδου έχει σχήμα  $1 \times n \times m$ , και σύμφωνα με την [126]  $m \times n \times 1$ .

Ο Πίνακας 9 παρουσιάζει την ακρίβεια, την ανάκληση, τη σταθμισμένη βαθμολογία F1 και την ορθότητα του συλλογικού μοντέλου PerceptionNet στα δεδομένα των δοκιμών PAMAP2, σε σύγκριση με τα συλλογικά μοντέλα των μεθόδων CNN και Convolutional LSTM. Το μοντέλο PerceptionNet πέτυχε σχεδόν 2% υψηλότερη ακρίβεια από το Convolutional LSTM και 4% υψηλότερη από την προσέγγιση CNN.

**Πίνακας 9: Σύγκριση αποτελεσμάτων στο σύνολο δεδομένων δοκιμής PAMAP2 της μεθόδου PerceptionNet με άλλα μοντέλα τεχνολογιών αιχμής [141].**

Method	Precision	Recall	F1-score	Accuracy
CNN [118]	85.51%	84.53%	84.57%	84.53%
Convolutional LSTM [126]	87.75%	86.78%	86.83%	86.78%
PerceptionNet	<b>89.76%</b>	<b>88.57%</b>	<b>88.74%</b>	<b>88.56%</b>



**Εικόνα 55: Ο πίνακας σύγχυσης του συλλογικού μοντέλου PerceptionNet για το σύνολο δεδομένων PAMAP2 [141].**

Επιπλέον, η Εικόνα 55 δείχνει τον πίνακα σύγχυσης του αναπτυχθέντος μοντέλου στα δεδομένα δοκιμής PAMAP2. Δεν αποτελεί έκπληξη το γεγονός ότι το μοντέλο δυσκολεύεται και πάλι να διακρίνει τη δραστηριότητα *στέκομαι* από τη δραστηριότητα *κάθομαι*. Οι συγγραφείς των ερευνών [146][147] ισχυρίζονται ότι αυτή η εσφαλμένη ταξινόμηση είναι συνηθισμένο πρόβλημα στην ΑΑΔ, και ότι ένα επιπλέον IMU προσαρμοσμένο στο μηρό θα αποτελούσε μία λύση. Τέλος, πρέπει να σημειωθεί ότι η κλάση *σιδερώνω* είχε πολύ υψηλή ανάκληση (0.9913), αλλά πολύ χαμηλή ακρίβεια (0.6930), εξαιτίας του μεγάλου αριθμού ψευδών θετικών ταξινομήσεων. Με άλλα λόγια, το μοντέλο τείνει εσφαλμένα να κατατάσσει πολλές δραστηριότητες στη συγκεκριμένη κλάση.



## 6 Περιπτώσεις χρήσης του Κοινωνικού Διαδικτύου των Πραγμάτων για τον έλεγχο των προτεινόμενων γνωσιακών μηχανισμών

Προκειμένου να δοκιμασθεί ο προσομοιωτής ASSIST και να ελεγχθεί αν οι SIoT Agents επωφελούνται από τις σχέσεις τους, κάνοντας χρήση των προαναφερθέντων γνωσιακών μηχανισμών, επιλέχθηκαν δύο περιπτώσεις χρήσης: α) Πολιτιστικής Κληρονομιάς, και β) Ευημερίας.

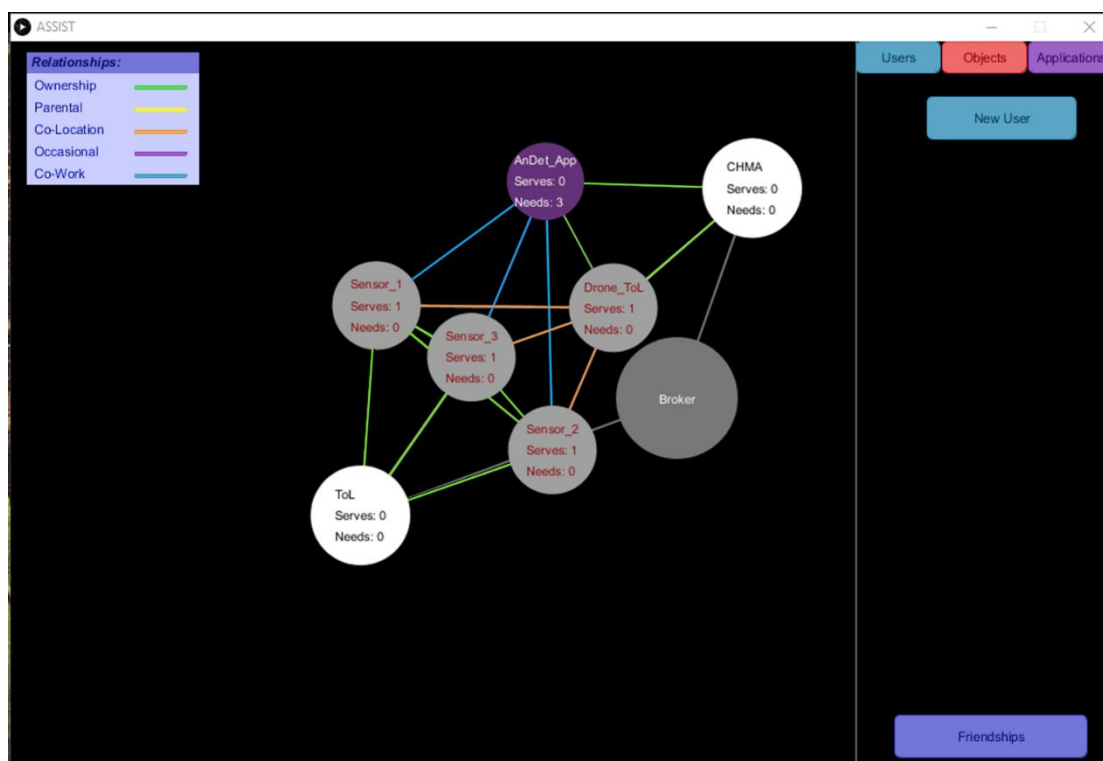
### 6.1 Περίπτωση χρήσης του Κοινωνικού Διαδικτύου των Πραγμάτων στην Πολιτιστική Κληρονομιά.

Η συγκεκριμένη περίπτωση χρήσης βασίζεται στην προστασία και συντήρηση των μνημείων Πολιτιστικής Κληρονομιάς. Τέτοιες περιπτώσεις χρήσης εξετάζονται στο ερευνητικό πρόγραμμα H2020 STORM [148], όπου Ασύρματα Δίκτυα Αισθητήρων (Wireless Sensor Networks; WSNs) έχουν τοποθετηθεί κοντά σε ιστορικά μνημεία, ώστε να μπορούν να παρακολουθήσουν ανθρωπογενείς και περιβαλλοντικούς παράγοντες όπως η υγρασία, ο ήχος, η πίεση κλπ. Αυτός ο τύπος υποδομής αυξάνει τη δυνατότητα ανίχνευσης επικίνδυνων συνθηκών σε συγκεκριμένες περιοχές ενδιαφέροντος. Ως αποτέλεσμα, ο συνδυασμός των WSNs με εργαλεία ανάλυσης δεδομένων επιτρέπει τη βελτίωση των μηχανισμών λήψης αποφάσεων, την ταχύτερη αντίδραση στην πρόληψη καταστροφών και την επιτάχυνση της αναστήλωσης τους.

Στην παρούσα διατριβή, επιλέχθηκε να εξεταστεί ένας αρχαιολογικός χώρος, όπου η εγκατάσταση των αισθητήρων απαιτεί ελάχιστη παρέμβαση στο περιβάλλον. Επομένως, προτιμήθηκε η χρήση συσκευών αισθητήρων μικρής κλίμακας, ικανών για επικοινωνία χαμηλής ισχύος. Ο αρχαιολογικός χώρος για την περίπτωση χρήσης είναι η *Οδός των Λεόντων στη Δήλο (Terrace of the Lions; ToL)*, η οποία βρίσκεται στο ελληνικό νησί της Δήλου. Σύμφωνα με την [149] μερικές από τις περιβαλλοντικές απειλές που αντιμετωπίζει ο χώρος είναι: η ατμοσφαιρική ρύπανση, η θερμοκρασία και οι σεισμικές δονήσεις. Συνεπώς, οι αισθητήρες που παρατηρούν αυτούς τους τρεις παράγοντες πρέπει να βρίσκονται κοντά στο μνημείο. Σύμφωνα με το σενάριο, η συλλογή των δεδομένων πραγματοποιείται από τους παραπάνω αισθητήρες, οι οποίοι στη συνέχεια πρόκειται να διαβιβάσουν τα δεδομένα τους σε Μη Επανδρωμένα Αεροσκάφη (Unmanned Aerial vehicles; UAVs). Τα συγκεκριμένα αεροσκάφη πετούν πάνω από τον επιλεγμένο τόπο με σκοπό την επιτήρηση και λειτουργούν ως πύλες δικτύου, υποστηρίζοντας τη μετάδοση δεδομένων από τους αισθητήρες. Τέλος, οι ροές δεδομένων μεταφέρονται σε μία διαγνωστική εφαρμογή, η οποία ανήκει σε μία *Αρχή Διαχείρισης Πολιτιστικής Κληρονομιάς (Cultural Heritage Management Authority; CHMA)* και θα ανιχνεύσει πιθανές ανωμαλίες. Στην

περίπτωση εντοπισμού μίας ανωμαλίας, η εφαρμογή στέλνει ειδοποιήσεις στους εμπειρογνώμονες.

Χρησιμοποιώντας τον προσομοιωτή ASSIST δημιουργήθηκαν δύο χρήστες, η CHMA και η ToL (Πίνακας 10). Η ToL διαθέτει τρεις αισθητήρες: *Sensor\_1*, *Sensor\_2* και *Sensor\_3*, οι οποίοι βρίσκονται κοντά στην τοποθεσία και έχουν ως υπηρεσίες: *Air\_Quality*, *Temperature* και *Earth\_Vibration* αντίστοιχα. Από την άλλη πλευρά, η CHMA είναι κάτοχος ενός UAV, του *Drone\_ToL*. Το *Drone\_ToL* βρίσκεται στην ίδια θέση με την ToL. Έτσι δημιουργήθηκε μία Σχέση Συνεγκατάστασης Αντικειμένων μεταξύ του *Drone\_ToL* και των αισθητήρων. Στη συνέχεια, εισήχθη στον προσομοιωτή ASSIST μία εφαρμογή ανίχνευσης ανωμαλιών που ονομάζεται *AnDet\_App*, η οποία έχει ως ανάγκες: *Air\_Quality*, *Temperature* και *Earth\_Vibration*, για να ανιχνεύει πιθανές επικίνδυνες καταστάσεις κοντά στο αρχαιολογικό μνημείο. Επιπλέον, η *AnDet\_App* ανήκει στην CHMA, έτσι, ο πράκτορας λογισμικού της δημιούργησε αυτόματα μία Σχέση Ιδιοκτησίας Αντικειμένου με τον πράκτορα λογισμικού του *Drone\_ToL*.



Εικόνα 56: Το GUI του ASSIST για την περίπτωση χρήσης της πολιτιστικής κληρονομιάς [19].

Κατά την πρώτη φάση εκπλήρωσης αναγκών, ο πράκτορας λογισμικού της *AnDet\_App* δεν κατάφερε να εκπληρώσει τις ανάγκες του, και, έπειτα, πέρασε στη δεύτερη φάση ανακάλυψης υπηρεσίας (Ενότητα 3.5.1). Ως αποτέλεσμα, ζήτησε από τους φίλους του να τον ενημερώσουν σχετικά με τις υπηρεσίες που προσφέρουν οι

Περιπτώσεις χρήσης του Κοινωνικού Διαδικτύου των Πραγμάτων για τον έλεγχο των προτεινόμενων γνωσιακών μηχανισμών

φίλοι τους. Κατά συνέπεια, ο πράκτορας λογισμικού ανακάλυψε τις υπηρεσίες που έψαχνε και έκανε αίτημα φιλίας στους πράκτορες λογισμικού των αισθητήρων της ToL. Τα στοιχεία συλλογισμού των πρακτόρων, οι οποίοι δέχθηκαν το αίτημα φιλίας, επεξεργάστηκαν το αίτημα κάνοντας χρήση σημασιολογικών κανόνων (Πίνακας 11) και, στη συνέχεια, το αποδέχτηκαν. Η Εικόνα 56 παρουσιάζει τις δημιουργούμενες κοινωνικές σχέσεις για την παρούσα περίπτωση χρήσης.

**Πίνακας 10: Οι οντότητες του SIoT οικοσυστήματος στο σενάριο χρήσης της πολιτιστικής κληρονομιάς [19].**

Όνομα	Τύπος	Ιδιοκτήτης	Φίλοι	Υπηρεσίες	Ανάγκες
ToL	User	-	Sensor_1, Sensor_2 Sensor_3	-	-
Sensor_1	Object	ToL	ToL, Sensor_2 Sensor_3, Drone_ToL	Air_Quality	-
Sensor_2	Object	ToL	ToL, Sensor_1 Sensor_3 Drone_ToL	Temperature	-
Sensor_3	Object	ToL	ToL, Sensor_1 Sensor_2 Drone_ToL	Earth_Vibration	-
CHMA	User	-	Drone_ToL	-	-
Drone_ToL	Object	CHMA	CHMA, Sensor_1 Sensor_2, Sensor_3	Surveillance	-
AnDet_App	App	CHMA	?	Anomaly_Detection	Air_Quality Temperature Earth_Vibration

**Πίνακας 11: Σημασιολογικοί κανόνες για τις γνωσιακές λειτουργίες των SIoT Agents [19].**

Όνομα Κανόνα	Αναπαράσταση Κανόνων κάνοντας χρήση του πλαισίου λογισμικού Jena
<i>makeFriend RequestToOwner</i>	(?me ssor:hasOwner ?f) -> (?me ssor:makeFriendRequest ?f)
<i>acceptRequests FromMyEntities</i>	(?me ssor:owns ?f) (?f ssor:makeFriendRequest ?me) -> (?me ssor:addFriend ?f)
<i>makeOwnership FriendRequest</i>	(?me ssor:hasOwner ?f) (?d ssor:hasOwner ?f) -> (?me ssor:makeFriendRequest ?d)
<i>deleteFriend</i>	(?me ssor:hasNumberOfFriends ?x) (?f rdf:type ssor:LeastImportantFriend) greaterThan(?x,n) -> (?me ssor:deleteFriend ?f)
<i>addFriend</i>	(?o ssor:makeFriendRequest ?me) (?o ssor:hasImportanceValue ?x) (?me ssor:hasFriend ?f) (?f ssor:hasImportanceValue ?y) greaterThan(?x,?y)

	-> (?me ssor:addFriend ?o)
<i>makeFriendRequest</i>	(?o ssor:hasImportanceValue ?x) (?me ssor:hasFriend ?f) (?f ssor:hasImportanceValue ?y) greaterThan(?x,?y) -> (?me ssor:makeFriendRequest ?o)
<i>subscribeToFriendsTopic</i>	(?me sons:needs ?n) (?n sons:isFulfilled "false") (?n sons:isSatisfiedBy ?s) (?me ssor:hasFriend ?a2) (?a2 sons:offers ?s) (?s sons:isRelatedTo ?t) -> (?me sons:subscribesTo ?t)
<i>makeFriendRequestToFoaF</i>	(?me sons:needs ?n) (?n sons:isFulfilled "false") (?n sons:isSatisfiedBy ?s) (?me ssor:hasFriend ?a2) (?a2 ssor:hasFriend ?a3) (?a3 sons:offers ?s) (?s sons:isRelatedTo ?t) -> (?me ssor:makeFoaFFriendRequest ?a3)

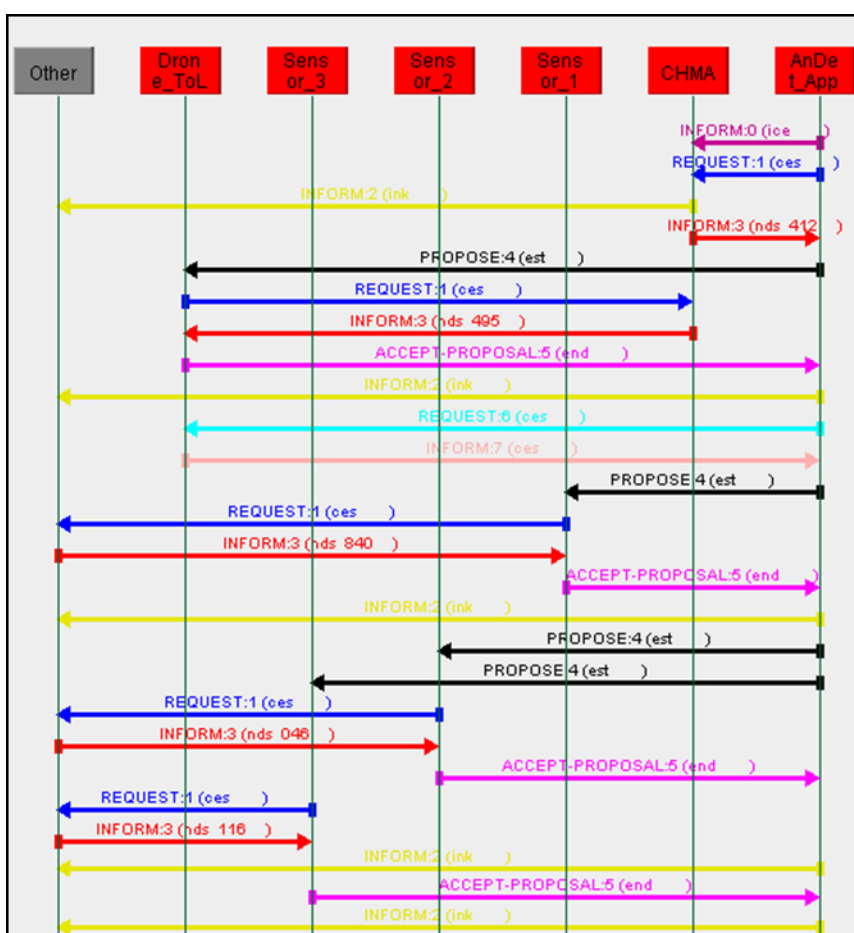
Οι σημασιολογικοί κανόνες που εισήχθησαν (Πίνακας 11) αναγκάζουν τους πράκτορες λογισμικού να αποφασίζουν ανάλογα με τον αριθμό των φίλων που έχουν, την αξία τους (*Importance*), και τις ανεκπλήρωτες ανάγκες τους. Οι δύο πρώτοι κανόνες, που παρουσιάζονται στον Πίνακας 11, αποσκοπούν στη σύνδεση μίας έξυπνης οντότητας με τον ιδιοκτήτη της. Συγκεκριμένα, ο κανόνας *makeFriendRequestToOwner* δίνει εντολή στον πράκτορα λογισμικού μίας συσκευής ή μίας εφαρμογής να στείλει αίτημα φιλίας στον ιδιοκτήτη της και ο κανόνας *acceptRequestsFromMyEntities* αναγκάζει τον πράκτορα λογισμικού του χρήστη να αποδέχεται αιτήματα φιλίας από οντότητες που του ανήκουν. Στη συνέχεια, βάσει του κανόνα *makeOwnershipFriendRequest* μία οντότητα στέλνει αίτημα φιλίας στις υπόλοιπες συσκευές του χρήστη (Σχέση Ιδιοκτησίας Αντικειμένου).

Ο κανόνας *deleteFriend* ενεργοποιείται όταν ένας πράκτορας λογισμικού έχει περισσότερους φίλους από ένα όριο *n*, οπότε επιλέγει να διαγράψει το φίλο του με τη χαμηλότερη τιμή *Importance* (*LeastImportantFriend*). Η τιμή *Importance* στο συγκεκριμένο σενάριο ήταν ένας τυχαίος αριθμός ή ένας αριθμός που έχει εισαγάγει ο χρήστης του ASSIST. Ο κανόνας *addFriend* οδηγεί στο να χαρακτηρισθούν ως φίλοι οι οντοτήτες που έχουν υποβάλει αίτημα φιλίας στον πράκτορα λογισμικού, εφόσον αυτές έχουν υψηλό δείκτη σημασίας. Επιπλέον, οι κανόνες *makeFriendRequest* και *makeFriendRequestToFoaF* αναγκάζουν έναν πράκτορα λογισμικού να προωθήσει αίτημα φιλίας σε «αγνώστους» κόμβους και σε φίλους των φίλων του, σύμφωνα με τις τιμές *Importance* και τις υπηρεσίες που προσφέρουν αντίστοιχα. Τέλος, ο κανόνας *subscribeToFriendsTopic*, συσχετίζει τις ανάγκες του πράκτορα λογισμικού με τις υπηρεσίες των φίλων του και τον βοηθά να εγγραφεί στα αντίστοιχα θέματα που τον ενδιαφέρουν.

Η Εικόνα 57 απεικονίζει το GUI του πράκτορα λογισμικού Sniffer της πλατφόρμας JADE, ο οποίος έχει καταγράψει όλα τα μηνύματα που απεστάλησαν μεταξύ των πρακτόρων στην περίπτωση χρήσης της πολιτιστικής κληρονομιάς, μετά τη



δημιουργία του πράκτορα *AnDet\_App*. Πρέπει να σημειωθεί, ότι ο μέγιστος αριθμός φίλων που μπορούσε να έχει μία οντότητα ήταν δέκα (δηλ., υψηλότερος από τον αριθμό των τελικών κοινωνικών δεσμών τους). Ως αποτέλεσμα, καμία από τις έξυπνες οντότητες δεν είχε περισσότερους φίλους από το καθορισμένο όριο και οι κανόνες που είχαν ως είσοδο το δείκτη σημασίας και το συγκεκριμένο όριο, δεν ενεργοποιήθηκαν. Στο ίδιο σενάριο έχοντας ορίσει το 3 ως μέγιστο αριθμό πιθανών φίλων για τους αισθητήρες, οι δύο αισθητήρες από τους τρεις (*Sensor\_1*, *Sensor\_3*) δεν δέχτηκαν το αίτημα φιλίας του πράκτορα *AnDet\_App*, εξαιτίας του ότι είχε χαμηλότερη τιμή *Importance* από τους υπάρχοντες φίλους τους και αρνήθηκαν να του προσφέρουν τις υπηρεσίες που χρειαζόταν.



Εικόνα 57: Το GUI του JADE sniffer agent που παρουσιάζει τα μηνύματα που ανταλλάξαν οι *IoT agents* στην περίπτωση χρήσης της πολιτιστικής κληρονομιάς, μετά τη δημιουργία του πράκτορα *AnDet\_App* [19].

Θα πρέπει να σημειωθεί ότι, ο αριθμός των ανταλλασσόμενων μηνυμάτων *FIPA* κατά τη φάση της ανακάλυψης υπηρεσίας εξαρτάται επίσης από τους προκαθορισμένους σημασιολογικούς κανόνες. Προκειμένου να εξετασθούν διαφορετικές προσεγγίσεις στην ανακάλυψη υπηρεσιών χρησιμοποιώντας τον

προσομοιωτή ASSIST, δημιουργήθηκαν δύο κανόνες: α) ο *flooding* και β) ο *naiveInfluencer* (Πίνακας 12). Ο πρώτος αναγκάζει τον SIoT Agent να ρωτήσει όλους του γειτονικούς κόμβους για τις υπηρεσίες των φίλων τους, ενώ ο τελευταίος υποχρεώνει τον πράκτορα λογισμικού να στείλει ένα μήνυμα πρώτα στον *MaxInfluencer* [150] των φίλων του. Ο *MaxInfluencer* μπορεί εύκολα να εντοπισθεί από τον SIoT Agent χρησιμοποιώντας ένα ερώτημα SPARQL.

Για τη μέτρηση του αριθμού των ανταλλασσόμενων μηνυμάτων, με βάση τους προαναφερθέντες κανόνες, δημιουργήθηκε ένας SIoT Agent, ο ιδιοκτήτης του οποίου είχε άλλες 5 έξυπνες οντότητες υπό τον έλεγχό του. Οι υποθέσεις για τους φίλους των φίλων αυτών των έξυπνων οντοτήτων ήταν:

1. όλοι οι φίλοι ενός πράκτορα λογισμικού είναι τυχαίοι φίλοι (δηλ., οι γείτονες ενός κόμβου δεν είναι συνδεδεμένοι μεταξύ τους) και δεν είναι φίλοι κλίκας,
2. το δίκτυο είναι ιεραρχικό (δηλ., ο αριθμός των φίλων του κάθε κόμβου δεν είναι ίσος για όλους τους κόμβους),
3. ο κόμβος που προσφέρει την υπηρεσία που χρειάζεται ο πράκτορας του SIoT έχει ελάχιστο μήκος διαδρομής, ίσο με 3 και
4. η πιθανότητα ένας κόμβος να παράγει την επιθυμητή υπηρεσία είναι ομοιόμορφα κατανεμημένη.

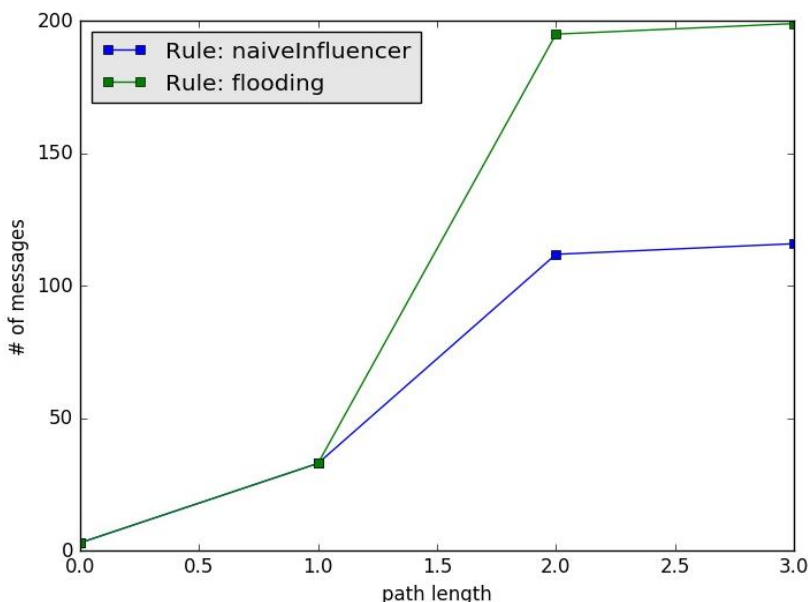
Η Εικόνα 58 δείχνει τον αριθμό των ανταλλασσόμενων μηνυμάτων κατά την ανακάλυψη υπηρεσίας σε σχέση με το μήκος διαδρομής, στην περίπτωση που ενεργοποιείται ο κανόνας *naiveInfluencer* ή ο κανόνας *flooding*. Όταν είναι ενεργοποιημένος ο κανόνας *flooding*, ο αριθμός των ανταλλασσόμενων μηνυμάτων αυξάνεται εκθετικά  $O(b^m)$  σε κάθε βήμα, ενώ στην περίπτωση της χρήσης του κανόνα *naiveInfluencer* αυξάνεται κατά  $O((b/2)^m)$ . Όπου  $b$  είναι ο αριθμός των κόμβων, και  $m$  το μήκος της διαδρομής (path length).

**Πίνακας 12: Σημασιολογικοί κανόνες για την ανακάλυψη υπηρεσιών.**

<i>flooding</i>	(?me sons:needs ?n) (?n sons:isFulfilled "false") (?me ssor:hasFriend ?a2) (?a2 ssor:hasFriend ?a3) -> (?me ssor:requestFoaFServices ?a2)
<i>naiveInfluencer</i>	(?me sons:needs ?n) (?n sons:isFulfilled "false") (?me ssor:hasFriend ?a2) (?a2 ssor:hasFriend ?a3) (?a2 rdf:type ssor:MaxInfluencer) -> (?me ssor:requestFoaFServices ?a2)

Πρέπει να σημειωθεί, ότι τα μηνύματα που αποστέλλονται στον Broker Agent του ASSIST δεν ελήφθησαν υπόψη, επειδή αυτά τα μηνύματα αποτελούν μέρος του οικοσυστήματος ASSIST (δηλ., για το σχεδιασμό των ακμών) και όχι του SIoT.

Επιπλέον, για την εξαγωγή των αποτελεσμάτων αναπαράχθηκε το σενάριο 100 φορές, επιλέγοντας κάθε φορά έναν τυχαίο κόμβο για την προσφορά της προς ανακάλυψη υπηρεσίας (πληρώντας σε κάθε περίπτωση τις προαναφερθείσες υποθέσεις).



Εικόνα 58: Ο αριθμός των μηνυμάτων FIPA που ανταλλάχθηκαν κατά τη διαδικασία ανακάλυψης υπηρεσίας, κάνοντας χρήση του κανόνα naiveInfluencer (μπλε χρώμα) ή του κανόνα flooding (πράσινο χρώμα).

## 6.2 Περίπτωση χρήσης του Κοινωνικού Διαδικτύου των Πραγμάτων στην Ευημερία.

Ο πολλαπλασιασμός των φορετών συσκευών (π.χ., Fitbit Charge HR [151]), οι οποίες είναι ικανές να μετρούν σε πραγματικό χρόνο χαρακτηριστικά που σχετίζονται με την υγεία και την ευημερία (π.χ., θερμίδες που καίγονται από το χρήστη), έχει αυξήσει το ενδιαφέρον πολλών ερευνητών για την ανάπτυξη εφαρμογών υγείας και ευημερίας [152][153]. Επιπλέον, υπάρχουν πολλές εμπορικές εφαρμογές (π.χ., Nike + [154], Adidas miCoach [155]) που υπόσχονται εξατομικευμένη προπόνηση στους χρήστες τους. Ωστόσο, οι υπάρχουσες εφαρμογές δεν λαμβάνουν υπόψη δεδομένα υγείας ή ευημερίας που παρακολουθούνται από συσκευές οι οποίες δεν είναι συμβατές με αυτές (π.χ., η εφαρμογή miCoach δεν επικοινωνεί με μία εφαρμογή καιρού ή με μία ηλεκτρονική ζυγαριά Withings [156]) και σε πολλές περιπτώσεις, ο χρήστης πρέπει να εισάγει χειροκίνητα τα δεδομένα που σχετίζονται με την κατάσταση της υγείας του (π.χ., δείκτης μάζας σώματος).

Με βάση το αυξανόμενο ενδιαφέρον για τη δημιουργία εφαρμογών υγείας και ευημερίας που κάνουν χρήση των δυνατοτήτων των φορετών συσκευών, επιλέχθηκε

ένα σενάριο ευημερίας με σκοπό τον έλεγχο των προτεινόμενων γνωσιακών μηχανισμών στο Κοινωνικό Διαδίκτυο των Πραγμάτων. Στην παρούσα περίπτωση χρήσης, ένα παντρεμένο ζευγάρι (ο *John* και η *Mary*), ζουν μαζί και μοιράζονται μερικά έξυπνα αντικείμενα. Τα αντικείμενα και οι εφαρμογές μπορούν παρέχουν πληροφορίες σε περισσότερους από έναν χρήστες, αλλά καθένα από αυτά έχει μόνο έναν κάτοχο. Ο *John* έχει στην κατοχή του μία φορητή συσκευή εντοπισμού δραστηριότητας (*activityTracker*), μία εφαρμογή καιρού (*weatherApp*), μία έξυπνη λάμπα (*lamp*) και έναν εβδομαδιαίο χρονοπρογραμματιστή (*weekScheduler*). Η *Mary* διαθέτει μία έξυπνη ηλεκτρονική ζυγαριά (*bodyScale*) και ένα κλιματιστικό (*airConditioner*). Ο Πίνακας 13 παρουσιάζει όλες τις υπάρχουσες έξυπνες οντότητες μαζί με τις ανάγκες και τις υπηρεσίες τους για τη συγκεκριμένη περίπτωση χρήσης.

**Πίνακας 13: Οι οντότητες του IoT στο σενάριο χρήσης της ευημερίας.**

Όνομα	Τύπος	Ιδιοκτήτης	Φίλοι	Υπηρεσίες	Ανάγκες	Εκδόσεις	Συνδρομές
John	User	-	[lamp, Mary, bodyScale, activityTracker, weekScheduler, weatherApp]	[location]	[workoutSchedule, bodyWeight Measuring]	[location]	[bodyWeight Measuring]
lamp	Object	John	[John, activityTracker, weekScheduler, weatherApp, bodyScale]	[lighting]	[luminosity]	[lighting]	[-]
activityTracker	Object	John	[John, lamp, weekScheduler, weatherApp]	[triaxialAccerelation, triaxialRotation, heartRateMeasuring]	[-]	[triaxialAccerelation, triaxialRotation, heartRateMeasuring]	[-]
weekScheduler	App	John	[John, activityTracker, weatherApp, lamp]	[schedule]	[-]	[schedule]	[-]
weatherApp	App	John	[John, activityTracker, scheduler, weatherApp]	[weather Conditons]	[location]	[weatherConditions]	[location]
Mary	Object	-	[John, bodyScale, airConditioner]	-	-	-	-
airConditoner	App	Mary	[Mary, bodyScale]	[cooling, heating]	[temperature]	[cooling, heating]	[-]
bodyScale	User	Mary	[Mary, John, airConditioner]	-	-	-	-

Προκειμένου να εξεταστούν οι προτεινόμενες γνωσιακές λειτουργίες του IoT Agent και ο κύκλος ζωής του, δημιουργήθηκε ένας κατάλογος σημασιολογικών κανόνων, οι οποίοι παρουσιάζονται σε μορφή Jena Rules στον Πίνακα 11 (Ενότητα 6.1). Έπειτα, χρησιμοποιώντας τον ASSIST, προσομοιώθηκε η προαναφερθείσα περίπτωση χρήσης για το IoT, και δημιουργήθηκε μία νέα εφαρμογή που ονομάζεται *fitnessApp* (Πίνακας 14), για να εξεταστεί η συμπεριφορά της. Στο παρόν σενάριο, ο

*John* σε μία προσπάθειά του να εκπληρώσει την ανάγκη του να ασκηθεί, έχει κατεβάσει αυτήν την εφαρμογή. Συγκεκριμένα, η *fitnessApp* ανήκει στον *John* και έχει ως υπηρεσία: *workoutSchedule*, ενώ ως ανάγκες: *triaxialAcceleration*, *triaxialRotation*, *heartRateMeasuring*, *weatherConditions*, *schedule*, και *bodyWeightMeasuring*.

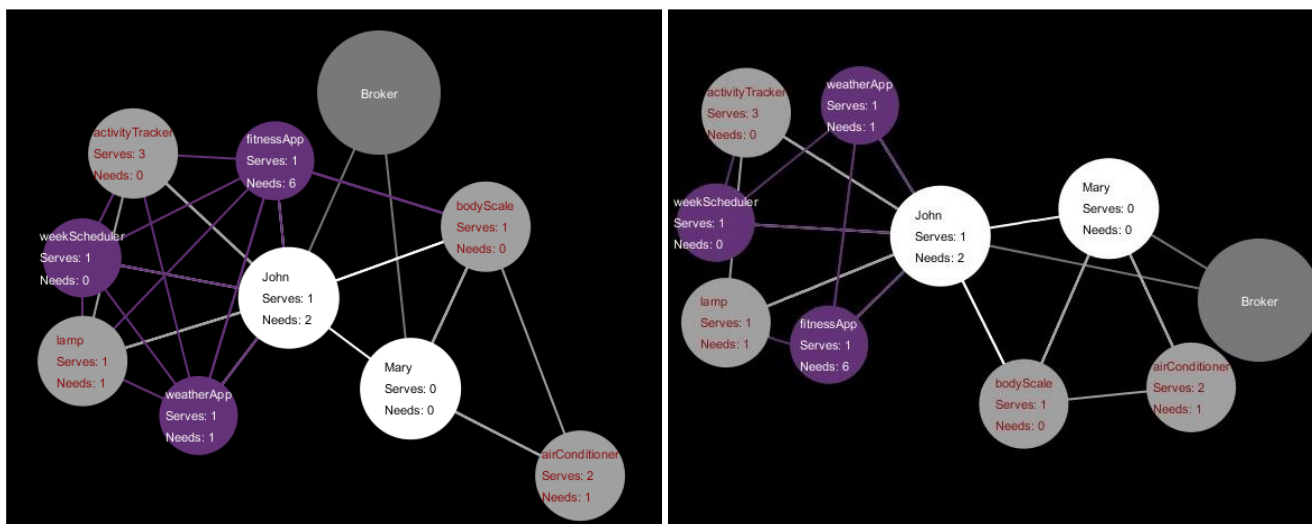
**Πίνακας 14: Οι ανάγκες και οι υπηρεσίες της οντότητας *fitnessApp*.**

Όνομα	Τύπος	Ιδιοκτήτης	Φίλοι	Υπηρεσίες	Ανάγκες	Εκδόσεις	Συνδρομές
<i>fitnessApp</i>	App	<i>John</i>	-	[ <i>workoutSchedule</i> ]	[ <i>triaxialAcceleration</i> , <i>triaxialRotation</i> , <i>heartRateMeasuring</i> , <i>weatherConditions</i> , <i>schedule</i> , <i>bodyWeightMeasuring</i> ]	-	-

Αυτές οι ανάγκες θα βοηθήσουν τη *fitnessApp* να δημιουργήσει ένα βελτιστοποιημένο πρόγραμμα τζόκινγκ (π.χ., επιλογή ημέρας για τρέξιμο με χαμηλή πιθανότητα βροχής) για τον *John*. Μετά την υλοποίηση της πρώτης φάσης, ο πράκτορας της *fitnessApp* έχει ως φίλους ιδιοκτησίας τους πράκτορες: *John*, *weatherApp*, *activityTracker*, *lamp* και *weekScheduler*. Χρησιμοποιώντας τον κανόνα *subscribeToFriendsTopic*, εγγράφεται στα θέματα *triaxialAcceleration*, *triaxialRotation*, *heartRateMeasuring* που δημοσιεύει η *activityTracker* και στα θέματα *weatherConditions*, *schedule* που παράγονται από τη *weatherApp* και τον *weekScheduler* αντίστοιχα. Ωστόσο, η ανάγκη *bodyWeightMeasuring*, παραμένει ανεκπλήρωτη. Κατά συνέπεια, ο πράκτορας λογισμικού ζητά από τους φίλους του να του στείλουν μία λίστα που περιέχει τους φίλους τους και τις υπηρεσίες που προσφέρουν. Ο φίλος του *airConditioner*, τον ενημερώνει ότι έχει έναν φίλο, τη *bodyScale*, που παράγει την υπηρεσία *BodyWeMeasuring*. Αυτό το γεγονός ενεργοποιεί τον κανόνα *makeFriendRequestToFoaF* για αποστολή αιτήματος φιλίας στη *bodyscale*, το οποίο έγινε δεκτό και ενεργοποιήθηκε ο κανόνας *subscribeToFriendsTopic*.

Στα αριστερά της Εικόνα 59 παρουσιάζεται ο δημιουργημένος κόσμος του *SlIoT*. Θα πρέπει να σημειωθεί, ότι για τη συγκεκριμένη περίπτωση χρήσης χρησιμοποιήθηκε η έκδοση Beta του προσομοιωτή *ASSIST* [44] και όχι η τελική [19] και για αυτό το λόγο λείπουν τα χρώματα που παρουσιάζουν στο χρήστη το είδος των δημιουργημένων σχέσεων φιλίας. Επιπλέον, το όριο που επιλέχθηκε για το μέγιστο αριθμό φίλων ήταν ίσο με 10. Η εκτέλεση του το ίδιου σεναρίου και η χρήση των ίδιων σημασιολογικών κανόνων με ένα μέγιστο όριο φίλων ίσο με 3, οδήγησε σε ένα πολύ διαφορετικό κοινωνικό γράφημα, το οποίο φαίνεται στο δεξί μέρος της Εικόνα 59. Αυτό οφείλεται στην ενεργοποίηση του κανόνα *deleteFriend*, σύμφωνα με τον οποίο εάν ένα αντικείμενο έχει περισσότερους φίλους από ένα όριο, πρέπει να

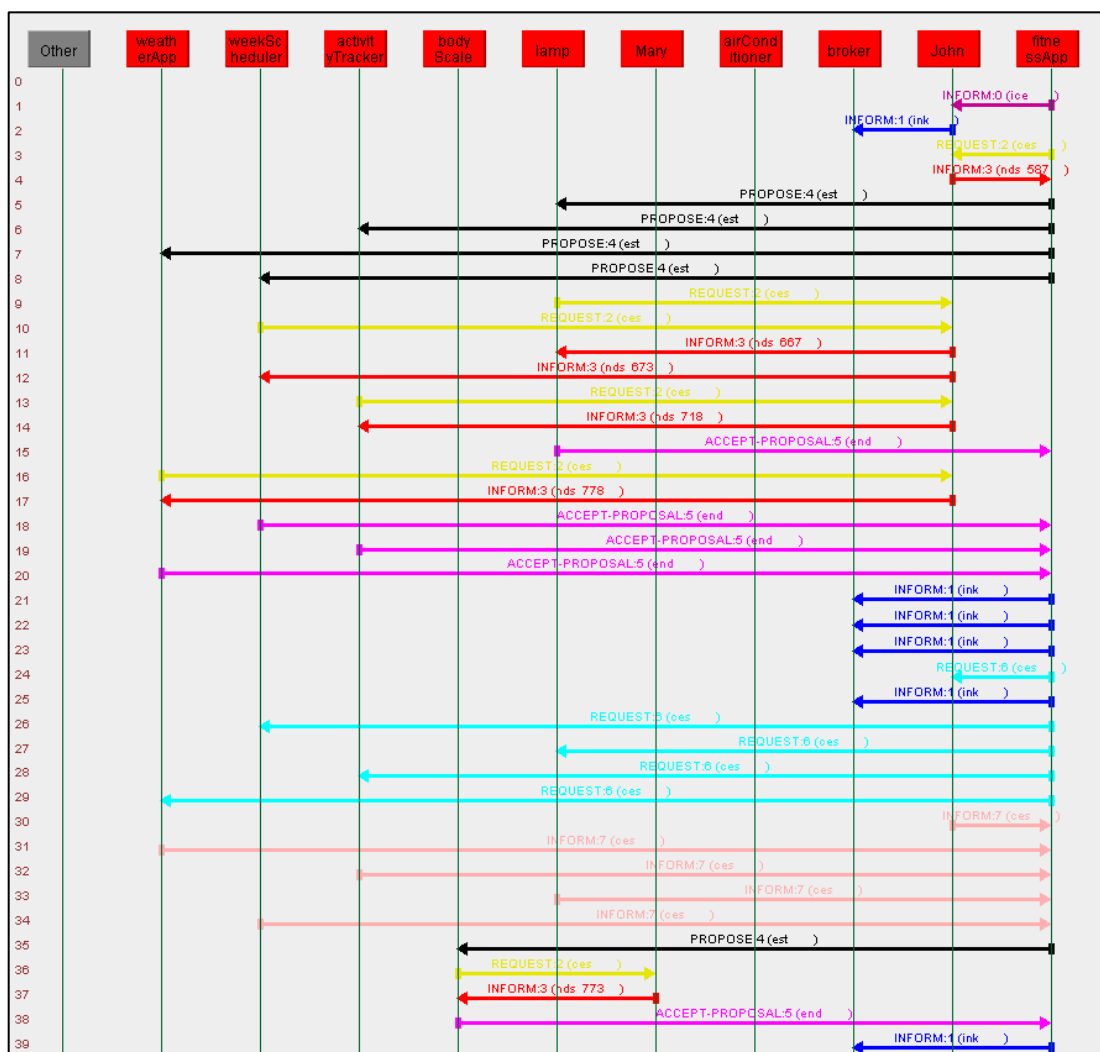
διαγράψει τις λιγότερο σημαντικές (με βάση την αξία τους) φίλιες του. Σε αυτή την περίπτωση, η *fitnessApp* δεν κατόρθωσε να καλύψει όλες τις ανάγκες της. Ωστόσο, αυτό δεν σημαίνει ότι ο πράκτορας λογισμικού της δεν θα συνεργαστεί καθόλου με τους άλλους πράκτορες, αλλά ότι οι υπηρεσίες που θα προσφέρει δεν θα είναι οι βέλτιστες.



Εικόνα 59: Δύο screenshots που παρουσιάζουν το οικοσύστημα του SIoT για την περίπτωση χρήσης της ευημερίας. Στο αριστερό, το όριο του μέγιστου αριθμού σχέσεων ισούται με δέκα, ενώ στο δεξί, με τρία.

Η Εικόνα 60 δείχνει το γραφικό περιβάλλον του πράκτορα Sniffer της πλατφόρμας JADE, ο οποίος είναι υπεύθυνος για την καταγραφή των μηνυμάτων που ανταλλάσσουν οι πράκτορες, έχοντας όριο μέγιστου αριθμού σχέσεων ίσο με 10. Στα πρώτα τέσσερα μηνύματα ο πράκτορας λογισμικού της *fitnessApp* ενημερώνει τον πράκτορα λογισμικού του *John* και τον Broker Agent για την ύπαρξή του και ζητά από τον πρώτο να του αποστείλει τα μοναδικά αναγνωριστικά των έξυπνων οντοτήτων που κατέχει. Αφού παραλάβει τις απαντήσεις του Πράκτορα Χρήστη, στέλνει ένα αίτημα φιλίας σε όλες τις οντότητες του χρήστη *John* (μηνύματα 5-8). Μόλις δέχτηκαν το αίτημα δημιουργίας Σχέσης Ιδιοκτησίας Αντικειμένου οι συγκεκριμένες οντότητες (*activityTracker*, *weatherApp*, *lamp* και *weekScheduler*), ζήτησαν από τον πράκτορα λογισμικού του *John* να τους στείλει την ανανεωμένη λίστα με τις έξυπνες συσκευές που κατέχει, και, στη συνέχεια, ενημέρωσαν το δικό τους κατάλογο. Ύστερα από την επιβεβαίωση ότι η *fitnessApp* ανήκει στον *John*, έστειλαν ένα μήνυμα αποδοχής αιτήματος φιλίας στον πράκτορα *fitnessApp*, το οποίο περιείχε τις υπηρεσίες που παρέχουν και τα URI των θεμάτων τους. Έπειτα, ο πράκτορας *fitnessApp* ενημέρωσε τον Broker Agent για τους δημιουργούμενους συνδέσμους και έγινε συνδρομητής στα θέματα που ικανοποιούν τις ανάγκες του. Επειδή δεν καλύφθηκαν όλες οι ανάγκες του στα μηνύματα 24, 26-29, ζήτησε από όλους τους φίλους του να του

στείλουν τις υπηρεσίες των φίλων τους και αυτοί του απάντησαν στα μηνύματα 30-34. Χρησιμοποιώντας το μηχανισμό πλημμύρας δικτύου, ο πράκτορας *fitnessApp* ανακάλυψε πως η υπηρεσία που επιδιώκει, προσφέρεται από τον πράκτορα λογισμικού της *bodyScale* και έστειλε ένα αίτημα φιλίας σε αυτόν. Τέλος, ο πράκτορας λογισμικού της *bodyScale* αποδέχθηκε το αίτημά του και ο πράκτορας *fitnessApp* κατάφερε να εκπληρώσει όλες τις ανάγκες του.



Εικόνα 60: Το GUI του JADE sniffer agent που παρουσιάζει τα μηνύματα που αντάλλαξαν οι IoT agents στην περίπτωση χρήσης της ευημερίας, μετά τη δημιουργία του *fitnessApp* agent.

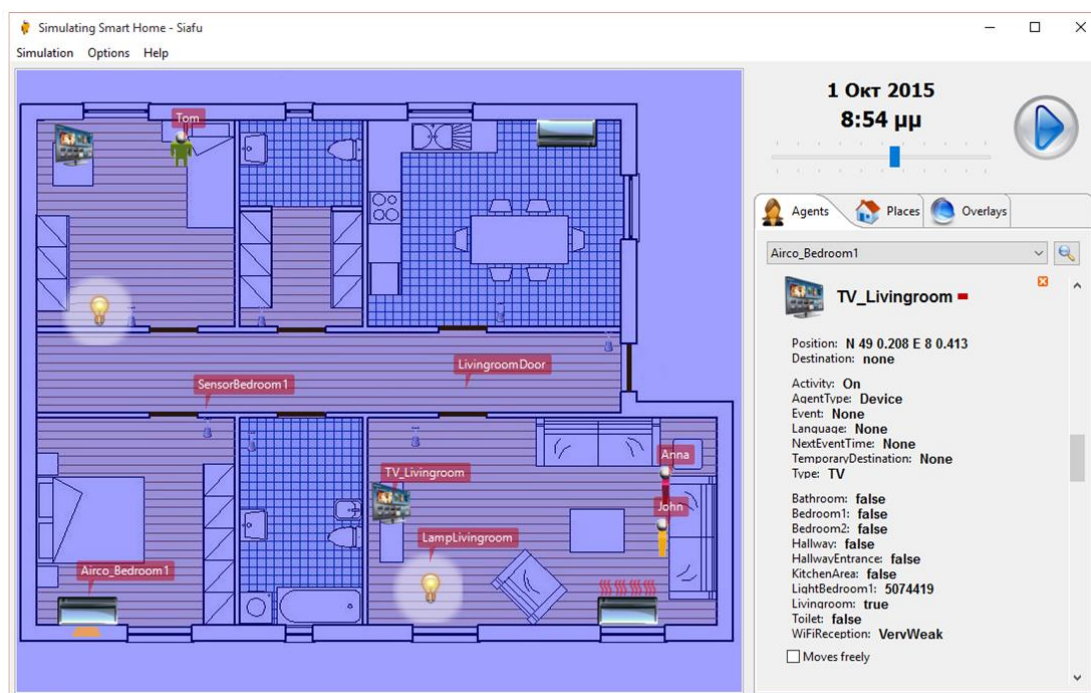
## 6.3 Περίπτωση χρήσης έξυπνου σπιτιού

### 6.3.1 Έλεγχος του μηχανισμού ανίχνευσης συγκρούσεων

Για να ελεγχθεί ο μηχανισμός ανίχνευσης συγκρούσεων κανόνων στην προτεινόμενη πλατφόρμα IoT, προσομοιώθηκε ένα οικοσύστημα. Ο προσομοιωμένος κόσμος λαμβάνει χώρα σε περιβάλλον έξυπνου σπιτιού. Ένα



ζευγάρι (*John Smith* και *Anna Smith*) ζει μαζί με το παιδί τους (*Tom*), σε ένα έξυπνο σπίτι. Το σπίτι χωρίζεται σε επτά δωμάτια (*kitchen*, *livingroom*, *hallway*, *bedroom1*, *bedroom2*, *bath* και *toilet*) που είναι εξοπλισμένα με πληθώρα έξυπνων συσκευών, όπως αισθητήρες παρουσίας, αισθητήρες θερμοκρασίας, αισθητήρες υγρασίας, ενεργοποιητές πόρτας, λαμπτήρες, τηλεοράσεις, κλιματιστικά και μία εφαρμογή εξοικονόμησης ενέργειας. Για τις συσκευές αυτές δημιουργήθηκαν τα αρχεία που περιέχουν τις καταστάσεις, τις υπηρεσίες, τις εντολές και τους κανόνες τους. Αυτά τα αρχεία είναι σε μορφή OWL, σχεδιάστηκαν στο εργαλείο protégé 4.3 και θεωρείται ότι έχουν δημιουργηθεί από τον κατασκευαστή. Οι οντολογίες των συσκευών ακολούθησαν την οντολογία κορμού (Core). Ένα 24ωρο παράδειγμα της καθημερινότητας των τριών κατοίκων και των συσκευών τους γράφτηκε σε ένα αρχείο csv.



Εικόνα 61: Το προσομοιωμένο περιβάλλον έξυπνου σπιτιού με χρήση του προσομοιωτή Sifa.

Για την προσομοίωση αυτού του σεναρίου, επιλέχθηκε ο προσομοιωτής Sifa [157]. Είναι ένας προσομοιωτής ανοιχτού κώδικα (open-source) που κάνει χρήση πρακτόρων για τη μοντελοποίηση της συμπεριφοράς ανθρώπων και αντικειμένων. Στο συγκεκριμένο εργαλείο αναπτύχθηκε, λοιπόν, ένας κόσμος που έκανε χρήση της προτεινόμενης SIoT πλατφόρμας (Εικόνα 61). Πριν από την ενεργοποίηση του προσομοιωτή, όλα τα στιγμιότυπα της περίπτωσης χρήσης και οι τοποθεσίες του σπιτιού εισήχθησαν στην οντολογία Core από την ΚΔΣ της Sem-SHUI. Κατά τη δημιουργία ενός έξυπνου αντικειμένου «γεννιόταν» και ο αντίστοιχος πράκτορας λογισμικού της συσκευής. Για την υλοποίηση των πρακτόρων λογισμικού



χρησιμοποιήθηκε και πάλι (όπως και στην περίπτωση του ASSIST) το πλαίσιο λογισμικού *JADE* [131], καθώς προσφέρει μία πλατφόρμα που απλοποιεί την υλοποίηση πολυπρακτορικών συστημάτων.

Ένας πράκτορας λογισμικού κατά την εκκίνησή του διαβάζει όλες τις λειτουργίες της συσκευής που αντιπροσωπεύει και χρησιμοποιώντας το μοτίβο Pub-Sub που προσφέρει το πρωτόκολλο MQTT, δημιουργεί τα αντίστοιχα θέματα (π.χ., το θέμα κανάλι για μία τηλεόραση). Ύστερα, ακολουθεί η ανάπτυξη σχέσεων φιλίας με τα άλλα υπάρχοντα αντικείμενα/ τις άλλες υπάρχουσες εφαρμογές και η συνδρομή των *SiOT Agents* στα θέματα που τους ενδιαφέρουν για να καλύψουν τις ανάγκες τους. Όπως αναφέρθηκε προηγουμένως, όταν μεταβληθεί η κατάσταση ενός αντικειμένου, ο πράκτορας λογισμικού του (αντικειμένου) ενημερώνει το θέμα του και όλοι οι συνδρομητές αυτού του θέματος λαμβάνουν ένα μήνυμα. Με βάση την ώρα που λαμβάνει χώρα ένα συμβάν, ο πράκτορας *SiOT* διαβάζει τους κανόνες του κατασκευαστή και τους κανόνες του χρήστη που είναι ενεργοί αυτή τη χρονική στιγμή, για να εξετάσει αν η μεταβολή αυτή τους ενεργοποιεί ή τους απενεργοποιεί, κάνοντας χρήση παραγωγικού συλλογισμού. Αν ο πράκτορας, τελικά, καταλήξει να αλλάξει την κατάστασή του, τότε ενημερώνει το αντίστοιχο θέμα του. Για λόγους απλοποίησης του προβλήματος, ορίστηκε η κάθε συσκευή να ενημερώνει την κατάστασή της κάθε 10 δευτερόλεπτα, στον κόσμο προσομοίωσης.

**Πίνακας 15: Σύνολο κανόνων που έχουν ως δράστη το κλιματιστικό που βρίσκεται στο καθιστικό.**

Δημιουργός	Ονομασία	Κώδικας (SWRL or Jena)	Χρονικό διάστημα λειτουργίας	Ανίχνευση σύγκρουσης
Manufacturer	StandByMode	AirCotioner(?a) ^ hasEPCCode(?a, "1234567.800023.500") ^ hasLocation(?a,?l) ^ TemperatureCommand (?c) ^ hasCommand(?a, ?c) ^ hasCommandValue(?c,?t) ^ hasTemperatureValue(?l, ?x) ^ equal(?t,?x) → hasCommandValue(?a, "StandBy")	00:00:00 – 23:59:00	-
User	OccupiedAircoOn	(?AircoL core:name "AicoL") ^ (?PresenceSensorL core:hasObservationValue "True") ^ (?AircoL core:hasStateValue "Off") → (?AircoL core:hasCommandValue "On")	08:00:00 – 23:59:00	False
User	EmptyAircoOff	(?AircoL core:name "AicoL") ^ (? PresenceSensorL core:hasObservationValue "False") ^ (?AircoL core:hasStateValue "On") → (?AircoL core:hasCommandValue "Off")	08:00:00 – 23:59:00	False
User	RaiseTempl	(?AircoL core:name "AicoL") ^ (? PresenceSensorL core:hasObservationValue "False") ^ (?TemperatureSensorL core:hasObservationValue ?t) ^ lessThan (?t, "20"^^ double) → (?AircoL core:hasCommandValue "On") ^ (?AircoL core:hasCommandValue "22"^^double)	22:00:00 – 01:00:00	False

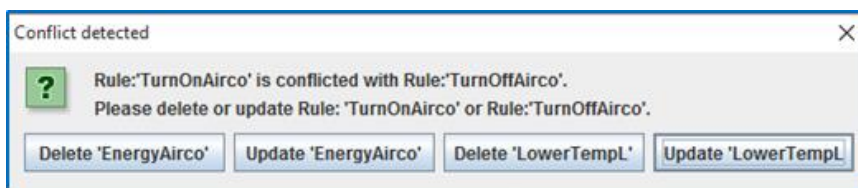
Περιπτώσεις χρήσης του Κοινωνικού Διαδικτύου των Πραγμάτων για τον έλεγχο των προτεινόμενων γνωσιακών μηχανισμών

User	LowerTempL	(?AircolL core:name "AircolL") ^ (? PresenceSensorL core:hasObservationValue "False") ^ (?TemperatureSensorL core:hasObservationValue ?t) ^ greaterThan (?t, "28"^^ double) → (?AircolL core:hasCommandValue "On") ^ (?AircolL core:hasCommandValue "25"^^double)	09:00:00 – 23:30:00	True
User	EnergyAircol	(?EnergyApplication core:name "EnergyApplication") ^ (?AircolL core:name "AircolL") ^ (?EnergyApplication core:hasEvergyValue ?e) ^ greaterThan (?e, "8000"^^ double) ^ (?AircolL core:hasStateValue "On") ^ → (?AircolL core:hasCommandValue "Off")	00:00:00 – 21:30:00	True

Κατά τη διάρκεια του χρόνου εκτέλεσης ο προσομοιωτής SifaFu αναλύει το αρχείο κειμένου που περιέχει την καθημερινότητα των κατοίκων και συνεπώς, οι Πράκτορες Χρήστη, οι Πράκτορες Συσκευών και οι Πράκτορες Εργασίας ξεκινούν να εκτελούν τις αντίστοιχες δραστηριότητες. Οι συσκευές δέχονται εντολές, επίσης, μέσω της ΚΚΣ της διεπαφής Sem-SHU, ενώ, όλες οι αλλαγές στην κατάσταση μίας οντότητας εμφανίζονται σε αυτή την καρτέλα.

Για να ελεγχθεί η άμεση εισαγωγή σημασιολογικών κανόνων στον κόσμο του προσομοιωτή SifaFu και η υπηρεσία ανίχνευσης συγκρούσεων της προτεινόμενης πλατφόρμας, δημιουργήθηκε ένα σύνολο κανόνων σε μορφή Jena που συνδυάστηκαν με τους κανόνες σε μορφή SWRL των κατασκευαστών (Πίνακας 15). Ο reasoner Pellet [45] επιλέχθηκε για να αναλύσει αυτούς τους κανόνες. Σημειώνεται, ότι ο προσομοιωτής SifaFu έχει την ικανότητα να γνωρίζει πού βρίσκονται οι πράκτορές του. Κάθε δωμάτιο του σπιτιού που είχε τουλάχιστον έναν πράκτορα χρήστη σε αυτό, χαρακτηρίζεται ως *Occupied* (κατειλημμένο) από τον προσομοιωτή και δημιουργείται ένα στιγμιότυπο αυτού του δωματίου στην οντολογία Core. Ως αποτέλεσμα, οι αισθητήρες παρουσίας άλλαξαν την τιμή παρατήρησής τους αν το δωμάτιο ήταν κατειλημμένο.

Ο Πίνακας 15 παρουσιάζει ένα σύνολο σημασιολογικών κανόνων IoT που έχουν σαν δράστη το κλιματιστικό που βρίσκεται στο σαλόνι του έξυπνου σπιτιού. Ενώ οι κανόνες των τεσσάρων πρώτων χρηστών δεν ήρθαν σε σύγκρουση με άλλους όταν εισήχθησαν στην ΚΔΚ, ο πέμπτος και ο τέταρτος ήταν αντικρουόμενοι. Ο λόγος είναι ότι έχουν αντίθετες εξόδους (ενεργοποίησε και απενεργοποίησε το κλιματιστικό *AircolL*). Αντίθετες συνέπειες έχουν, επίσης, και άλλοι κανόνες (*RaiseTempL* και *EnergyAircol*), αλλά λόγω διαφορετικών χρονικών περιόδων ενεργοποίησης ή επειδή ακολουθούν την “Αρχή μη-σύγκρουσης” (*OccupiedAircolOn* και *EmptyAircolOff*, *RaiseTempL*), δεν εντοπίστηκε σύγκρουση μεταξύ τους. Η Εικόνα 62 παρουσιάζει το μήνυμα προειδοποίησης που εμφανίζεται στο χρήστη μέσω της Sem-SHUI.



Εικόνα 62: Προειδοποιητικό μήνυμα κατά την εύρεση αντικρουόμενων κανόνων.

### 6.3.2 Μέτρηση χρόνου απόκρισης του στοιχείου συλλογισμού

Για τη μέτρηση του χρόνου απόκρισης του προτεινόμενου αποκεντριοποιημένου μηχανισμού συλλογισμού, έγινε χρήση του προσομοιωτή Siafu, όπως και του ASSIST. Καταγράφηκε, λοιπόν, η καθυστέρηση μεταξύ της λήψης των δεδομένων και της επιβολής των ενεργών κανόνων για την περίπτωση χρήσης του έξυπνου σπιτιού, εξετάζοντας 4 διαφορετικά σενάρια ως προς το σύνολο των ενεργών κανόνων ανά συσκευή. Για κάθε σενάριο εξετάστηκαν 2 διαφορετικοί μηχανισμοί: α) κεντριοποιημένος (centralized), και β) αποκεντριοποιημένος (decentralized). Στην περίπτωση του αποκεντριοποιημένου μηχανισμού εξετάστηκαν δύο υποπεριπτώσεις, όπου ο πράκτορας λογισμικού (δράστης) δεν είναι φίλος με τον επιθυμητό πράκτορα λογισμικού (παράμετρος), ώστε να καταναλώσει τα δεδομένα του. Στην πρώτη το μήκος διαδρομής είναι ίσο με 2 (FoaF request), ενώ στη δεύτερη ίσο με 3 (FoaFoaF request).

Τα πειράματα εκτελέστηκαν σε σταθμό εργασίας εξοπλισμένο με Intel Core i7-7700K 4.20GHz CPU, ο οποίος έχει RAM 16 gigabytes και διαθέτει 4 πυρήνες. Επιπλέον, στον Πίνακα 16: Παράμετροι πειράματος μέτρησης του χρόνου απόκρισης των μηχανισμών συλλογισμού παρουσιάζονται οι παράμετροι του συγκεκριμένου πειράματος.

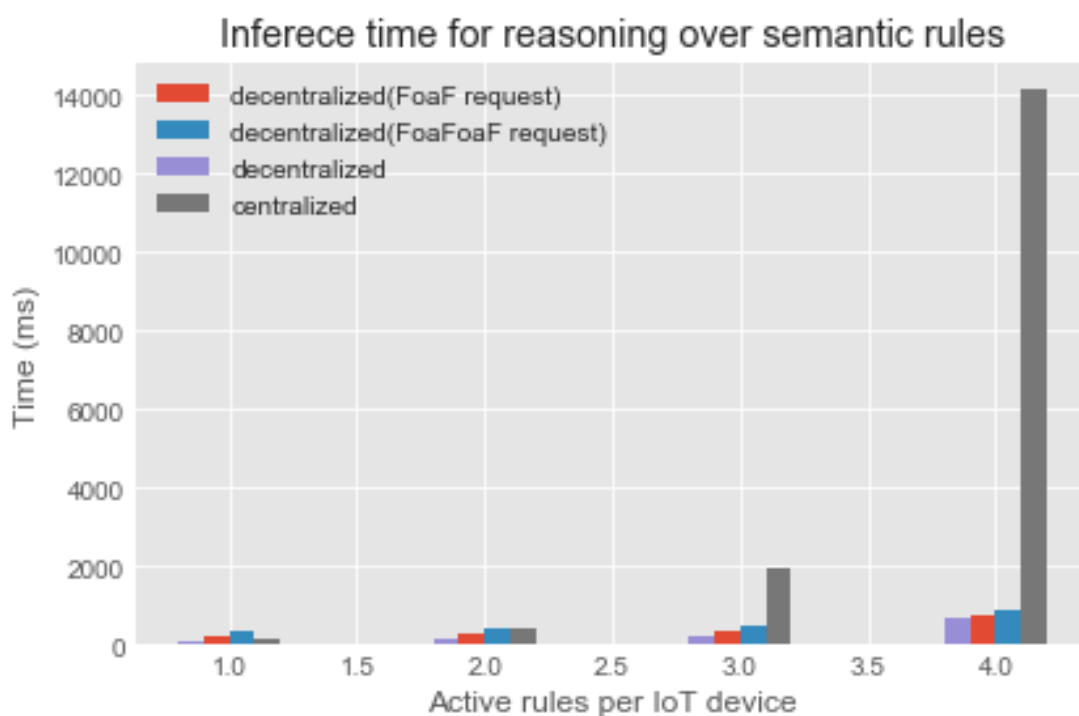
Πίνακας 16: Παράμετροι πειράματος μέτρησης του χρόνου απόκρισης των μηχανισμών συλλογισμού

Παράμετρος	Τιμές
Επαναλήψεις πειραμάτων	5
Αριθμός IoT συσκευών	40
Διατήρηση ιστορικών δεδομένων	Όχι
Μηχανισμός ανακάλυψης υπηρεσιών	Πλημμύρα
Μορφή μηνύματος	turtle
Καθυστέρηση δικτύου (MQTT μήνυμα)	~50 ms

## Περιπτώσεις χρήσης του Κοινωνικού Διαδικτύου των Πραγμάτων για τον έλεγχο των προτεινόμενων γνωσιακών μηχανισμών

Αριθμός τριπλετών ανά MQTT μήνυμα	8
Φίλοι ανά συσκευή	10

Στην Εικόνα 63 παρουσιάζονται τα αποτελέσματα που ανακτήθηκαν. Θα πρέπει να σημειωθεί ότι τα πειράματα διεξήχθησαν 5 φορές ανά περίπτωση και επιλέχθηκε η μέση τιμή της εκάστοτε καθυστέρησης, ενώ η καθυστέρηση του δικτύου (50 ms) δεν συνυπολογίστηκε. Τα αποτελέσματα αποδεικνύουν ότι προτεινόμενος μηχανισμός (decentralized) είναι ταχύτερος από τον κεντριοποιημένο (centralized), καθώς η καθυστέρηση του αυξάνεται γραμμικά  $O(mn)$ , όπου  $m$  ο χρόνος απόκρισης και  $n$  ο αριθμός των ενεργών κανόνων. Στην περίπτωση του κεντριοποιημένου παραγωγικού συλλογισμού η καθυστέρηση αυξάνεται πολυωνμικά  $O(m^n)$ . Για τις δύο περιπτώσεις όπου οι SIoT Agents των συσκευών πρέπει να ανακαλύψουν τους πράκτορες λογισμικού που προσφέρουν τα επιθυμητά προς συνδρομή θέματα, λόγω της καθυστέρησης για την ανακάλυψη της υπηρεσίας (~131 ms ανά μήκος διαδρομής), ο κεντριοποιημένος μηχανισμός εμφανίζει καλύτερη συμπεριφορά στην περίπτωση που αντιστοιχούσε ένας ενεργός κανόνας ανά συσκευή.



Εικόνα 63: Χρόνοι απόκρισης παραγωγικού συλλογισμού σε σχέση με τον αριθμό ενεργών κανόνων ανά συσκευή.

## 7 Συμπεράσματα και προτάσεις για μελλοντική έρευνα

Η παρούσα διατριβή έχει ως στόχο τη δημιουργία μίας αποκεντριοποιημένης πλατφόρμας για τη δυναμική διασύνδεση μεταξύ των οντοτήτων του Διαδικτύου των Πραγμάτων. Επιπλέον, η προτεινόμενη λύση αποσκοπεί στην εκμετάλλευση του μεγάλου όγκου δεδομένων που παράγουν οι αισθητήρες και οι έξυπνες συσκευές για τη δημιουργία εφαρμογών βασισμένων στην επίγνωση πλαισίου, και δυναμικών ροών εργασίας χωρίς την ανάγκη παρεμβολής των προγραμματιστών.

Υποστηρίζεται, λοιπόν, ότι μία έξυπνη οντότητα (αντικείμενο ή εφαρμογή) όχι μόνο θα πρέπει να εκτελεί τις εντολές του ιδιοκτήτη της, αλλά θα πρέπει, επίσης, να έχει τη δυνατότητα να λαμβάνει αποφάσεις βάσει των αναγκών του χρήστη και να δημιουργεί κοινωνικούς δεσμούς με τις υπόλοιπες έξυπνες οντότητες για να τις καλύψει. Ως αποτέλεσμα, κρίνεται απαραίτητη η ύπαρξη ενός γνωσιακού συστατικού στοιχείου λογισμικού. Μελετώντας το θεωρητικό υπόβαθρο, οι περιπτώσεις χρήσης του Κοινωνικού Διαδικτύου των Πραγμάτων και του Γνωσιακού Διαδικτύου των Πραγμάτων παρουσιάζουν αυτό το συστατικό στοιχείο λογισμικού ως μία έξυπνη οντότητα που βρίσκεται στο επίπεδο εφαρμογής του αντικειμένου, και ως μία οντότητα λήψης αποφάσεων πάνω από το στρώμα σημασιολογικής αναπαράστασης των δεδομένων, αντιστοίχως (Ενότητα 2.1.1.3). Για να επιτευχθεί η συγχώνευση αυτών των δύο προσεγγίσεων (δηλ., SIoT και CIoT), το έξυπνο συστατικό στοιχείο λογισμικού θα πρέπει να έχει τη δυνατότητα να λαμβάνει αποφάσεις όχι μόνο για τη διαχείριση στόχων, αλλά και για τη διαχείριση φιλίας.

Στην παρούσα διδακτορική διατριβή, λοιπόν, προτείνεται η χρήση γνωσιακών πρακτόρων λογισμικού με επίγνωση πλαισίου για την εκπροσώπηση των έξυπνων οντοτήτων, οι οποίες έχουν χαμηλή υπολογιστική ισχύ ή μνήμη, στο οικοσύστημα του SIoT. Τα ευφυή συστατικά στοιχεία του πράκτορα λογισμικού διαθέτουν απαγωγικό ή παραγωγικό συλλογισμό και ταξινομούνται σε τρία επίπεδα. Ο απαγωγικός συλλογισμός επιτυγχάνεται μέσω αλγορίθμων Βαθιάς Μάθησης, εφαρμόζεται στο χαμηλότερο επίπεδο, και αποσκοπεί στην εξαγωγή πληροφοριών πλαισίου από δεδομένα αισθητήρων κίνησης. Η εξαγόμενη πληροφορία μεταβιβάζεται στο δεύτερο επίπεδο και αναπαριστάται σημασιολογικά, διασφαλίζοντας τη διαλειτουργικότητα μεταξύ των πρακτόρων λογισμικού. Τέλος, ο παραγωγικός συλλογισμός επιτυγχάνεται μέσω σημασιολογικών κανόνων που έχει γράψει ο κατασκευαστής του αντικειμένου ή ο ιδιοκτήτης του. Τα αποτελέσματα αυτών προωθούνται σε σημασιολογικούς κανόνες για τη λήψη αποφάσεων όσον αφορά τη διαχείριση της κοινωνικής συμπεριφοράς και των στόχων των έξυπνων οντοτήτων.

### 7.1 Αξιολόγηση Αποτελεσμάτων

Η προτεινόμενη λύση αξιολογήθηκε σχετικά με τη δυνατότητα εξαγωγής πληροφοριών πλαισίου, τη δυνατότητα γνωσιακής διαχείρισης των κοινωνικών σχέσεων και την επίτευξη των στόχων των οντοτήτων του Διαδικτύου των Πάντων, και, τέλος, (σχετικά με) τη δυνατότητα εισαγωγής νέων κανόνων αυτοματισμού στα έξυπνα αντικείμενα.

Αρχικά, λοιπόν, μελετήθηκαν τα αποτελέσματα της επεξεργασίας των ακατέργαστων σημάτων κίνησης, προερχόμενα από εγκατεστημένους αισθητήρες σε φορητές/κινητές συσκευές, με στόχο την αναγνώριση ανθρώπινων δραστηριοτήτων. Πιο συγκεκριμένα, το Βαθύ Συνελικτικό Νευρωνικό μοντέλο που αναπτύχθηκε, εξετάστηκε όσον αφορά τα ποσοστά ορθότητας, ανάκλησης, ακρίβειας και σταθμισμένης βαθμολογίας F1. Με βάση τα δύο σύνολα δεδομένων που επιλέχθηκαν, κατάφερε να ξεπεράσει τα αντίστοιχα ποσοστά πρόσφατων ερευνητικών εργασιών που βασίσθηκαν σε τεχνολογίες αιχμής Μηχανικής Μάθησης (βασισμένα σε χαρακτηριστικά εξαχθέντα από ανθρώπους ή σε αλγορίθμους Βαθιάς Μάθησης). Επιπλέον, μελετώντας τον πίνακα σύγχυσης, προέκυψε ότι το αναπτυχθέν μοντέλο δυσκολεύεται να ξεχωρίσει αν ο χρήστης κάθεται ή είναι όρθιος, γεγονός που οφείλεται στην έλλειψη φορετών αισθητήρων προσαρμοσμένων στο μηρό του χρήστη.

Στη συνέχεια, μελετήθηκαν δύο περιπτώσεις χρήσης (ευημερίας και πολιτιστικής κληρονομιάς) για να ελεγχθεί κατά πόσο οι προτεινόμενοι γνωσιακοί πράκτορες λογισμικού, τα επιμέρους συστατικά τους στοιχεία λογισμικού, και ο προτεινόμενος κύκλος ζωής των έξυπνων αντικειμένων μπορούν να ανταπεξέλθουν στις απαιτήσεις τους. Συγκεκριμένα, εξετάσθηκε η δυνατότητα δυναμικής δημιουργίας ροών εργασίας και ανταλλαγής δεδομένων μεταξύ των SIoT Agents, κάνοντας χρήση του αναπτυχθέντος προσομοιωτή ASSIST. Εισαγάγοντας μία νέα συσκευή/εφαρμογή και ορίζοντας τις ανάγκες και τις υπηρεσίες της στο *αρχείο δήλωσης* η συσκευή/εφαρμογή, ακολουθώντας τα προτεινόμενα στάδια του κύκλου ζωής της, κατάφερε να επωφεληθεί του ήδη υπάρχοντος κοινωνικού δικτύου συνάπτοντας σχέσεις φιλίας και καλύπτοντας τις ανάγκες της. Για επιτευχθεί αυτό, ορίσθηκαν σημασιολογικοί κανόνες για την πλοήγηση στο κοινωνικό δίκτυο, τη διαχείριση των κοινωνικών σχέσεων και των στόχων τους, βασισμένοι στις αναπτυχθείσες οντολογίες SSOR και SONS.

Επιπλέον, εξετάσθηκε η δυνατότητα εισαγωγής νέων κανόνων αυτοματισμού, οι οποίοι οδηγούν στη δημιουργία νέων στόχων για τις συσκευές. Για το λόγο αυτό αναπτύχθηκε ένα σενάριο έξυπνου σπιτιού. Ο χρήστης εισήγαγε κανόνες αυτοματισμού, μέσω της διεπαφής Sem-SHUI που αναπτύχθηκε για αυτό το σκοπό, και στη συνέχεια αυτοί διανεμήθηκαν στους πράκτορες λογισμικού, οι οποίοι ήταν δράστες του κανόνα, από τον πράκτορα λογισμικού του χρήστη. Οι συγκεκριμένοι πράκτορες λογισμικού εξέτασαν επιτυχώς το κατά πόσο οι παραπάνω κανόνες είναι

πιθανώς αντικρουόμενοι, εκμεταλλεόμενοι την οντολογία Rule, η οποία σχεδιάστηκε για αυτό το σκοπό, και την προτεινόμενη “αρχή μη-σύγκρουσης”.

Τέλος, μελετήθηκαν οι χρόνοι απόκρισης του προτεινόμενου αποκεντριοποιημένου μηχανισμού συλλογισμού. Σύμφωνα με τα αποτελέσματα που ανακτήθηκαν ο προτεινόμενος μηχανισμός, ο οποίος βασίζεται σε γνωσιακούς πράκτορες λογισμικού, είχε ταχύτερους χρόνους απόκρισης. Ακόμα και στις περιπτώσεις που έπρεπε να δημιουργήσει τις απαραίτητες σχέσεις φιλίας, για να λαμβάνει τα δεδομένα που τον ενδιαφέρουν, όταν ο αριθμός των δεδομένων αυξανόταν ο χρόνος απόκρισης του κεντριοποιημένου συλλογισμού αυξανόταν πολυωνομικά, ενώ του προτεινόμενου μηχανισμού γραμμικά. Συνεπώς, σε περιβάλλοντα του IoT, όπου τα δεδομένα που παράγονται είναι μεγάλης κλίμακας, ο προτεινόμενος μηχανισμός καθίσταται ως ιδανικότερος.

Θα πρέπει να σημειωθεί ότι οι παραπάνω τεχνικές εξαγωγής και διαχείρισης της πληροφορίας πλαισίου διαθέτουν, εκτός από θετικά στοιχεία, και αρνητικά (Πίνακας 2). Ο Πίνακας 17 παρουσιάζει τις μεθόδους που επιλέχθηκαν και τους τρόπους αντιμετώπισης των μειονεκτημάτων τους. Με πλάγια γραφή παρουσιάζονται οι μέθοδοι αντιμετώπισης που δεν αναπτύχθηκαν στην παρούσα διδακτορική έρευνα, αλλά προτείνονται για μελλοντική έρευνα (Ενότητα 7.3.3).

**Πίνακας 17: Αντιμετώπιση προβλημάτων των επιλεχθεισών τεχνικών εξαγωγής και επεξεργασίας της πληροφορίας πλαισίου**

Τεχνικές	Αρνητικά	Αντιμετώπιση
<b>Επιβλεπόμενη Μάθηση</b>	<ul style="list-style-type: none"> <li>Απαιτείται σημαντική ποσότητα δεδομένων</li> </ul>	<ul style="list-style-type: none"> <li>Τα έξυπνα αντικείμενα και κυρίως οι αισθητήρες παράγουν χιλιάδες μετρήσεις μέσα σε ένα λεπτό</li> </ul>
(Αλγόριθμοι Βαθιάς Μάθησης)	<ul style="list-style-type: none"> <li>Κάθε στοιχείο δεδομένων πρέπει να μετατρέπεται σε αριθμητικές τιμές</li> <li>Η επιλογή των χαρακτηριστικών μπορεί να είναι απαιτητική</li> <li>Μπορεί να είναι πιο έντονη (επεξεργασία, αποθήκευση, χρόνος)</li> <li>Λιγότερο σημασιολογικά μοντέλα</li> <li>Απαιτούνται χαρακτηρισμένα δεδομένα εκπαίδευσης</li> <li>Τα μοντέλα μπορεί να είναι πολύπλοκα</li> <li>Δύσκολο να καταγράψουν την υπάρχουσα γνώση</li> </ul>	<ul style="list-style-type: none"> <li>Τα δεδομένα που παράγονται από τους αισθητήρες είναι εκφρασμένα σε αριθμητικές τιμές</li> <li>Οι αλγόριθμοι Βαθιάς Μάθησης έχουν τη δυνατότητα να εξαγάγουν χαρακτηριστικά αυτόματα</li> <li>Χρήση Μονάδας Επεξεργασίας Γραφικών (GPU) για την εκπαίδευση των μοντέλων</li> <li>Οι αλγόριθμοι Βαθιάς Μάθησης έχουν τη δυνατότητα σημασιολογικής αναπαράστασης των δεδομένων [158]</li> <li>Περιπτώσεις χρήσης Ημι-Επιβλεπόμενης Μάθησης (Semi-Supervised Learning)</li> <li>Ύπαρξη πολλών πλαισίων λογισμικού για την ανάπτυξη μοντέλων Βαθιάς Μάθησης (π.χ., Keras)</li> <li>Χρήση τεχνικών απομνημόνευσης, όπως Διαφοροποιήσιμος Νευρωνικός Υπολογιστής (Differentiable Neural Computer)</li> </ul>

<b>Οντολογίες</b>	<ul style="list-style-type: none"> <li>• Τα δεδομένα πρέπει να διαμορφώνονται σε συμβατή μορφή (π.χ., OWL, RDF)</li> <li>• Περιορισμένη αριθμητική συλλογιστική</li> <li>• Μπορεί να είναι πιο έντονη (επεξεργασία, αποθήκευση, χρόνος)</li> </ul>	<ul style="list-style-type: none"> <li>• Ανάπτυξη τεσσάρων οντολογικών μοντέλων (Core, Rule, SSOR, SONS)</li> <li>• Ο συλλογισμός πάνω στα αριθμητικά δεδομένα πραγματοποιείται σε προηγούμενο επίπεδο μέσω τεχνικών Βαθιάς Μάθησης</li> <li>• Αποκεντροποιημένος/η και κατανεμημένος/η συλλογισμός/αποθήκευση δεδομένων μέσω της χρήσης πρακτόρων λογισμικού</li> </ul>
(Λογική Βαθμού)	Πρώτου	
<b>Γλώσσες Κανόνων</b>	<ul style="list-style-type: none"> <li>• Πρέπει να ορίζονται με μη αυτόματο τρόπο</li> <li>• Μπορεί να είναι επιρρεπείς σε σφάλματα λόγω ανθρώπινου παράγοντα</li> <li>• Δεν υπάρχει επικύρωση ή έλεγχος ποιότητας</li> </ul>	<ul style="list-style-type: none"> <li>• Δημιουργία σημασιολογικών κανόνων μέσω της διεπαφής Sem-SHUI</li> <li>• Έλεγχος συντακτικών σφαλμάτων μέσω της διεπαφής Sem-SHUI</li> <li>• Ανίχνευση αντικρουόμενων κανόνων κατά τη δημιουργία τους μέσω της “Αρχής μη-σύγκρουσης”</li> </ul>
(Σημασιολογικοί Κανόνες)		

## 7.2 Υποθέσεις και Περιορισμοί

Για την αξιολόγηση της συγκεκριμένης διατριβής πραγματοποιήθηκαν κάποιες υποθέσεις με σκοπό την απλούστευσή της, καθώς και εντοπίστηκαν κάποιοι περιορισμοί σε σχέση με την εφαρμογή των επιλεχθεισών/αναπτυχθεισών τεχνικών σε προβλήματα «πραγματικού κόσμου». Συγκεκριμένα, οι υποθέσεις/περιορισμοί που έγιναν/εντοπίστηκαν είναι οι εξής:

- Τα αρχείο δήλωσης των συσκευών/εφαρμογών ακολουθεί την οντολογία Κορμού (Core ontology) και όχι κάποια οντολογία κατασκευαστή (Device ontology). Σε προβλήματα «πραγματικού κόσμου» οι οντολογίες των κατασκευαστών των αντικειμένων/εφαρμογών θα έπρεπε να αντιστοιχηθούν στην Core οντολογία.
- Το συστατικό στοιχείο λογισμικού *reasoner* βασίζει το συλλογισμό του πάνω στην τιμή *Importance*, η οποία ήταν ένας τυχαίος αριθμός στα σενάρια που παρουσιάσθηκαν. Για τον ορισμό της συγκεκριμένης τιμής θα πρέπει να ληφθούν υπόψιν αρκετοί παράμετροι, οι οποίοι ορίζονται στην επόμενη ενότητα.
- Για ανάπτυξη των μοντέλων χρησιμοποιήθηκαν σήματα προερχόμενα από τους ίδιους αισθητήρες. Όμως, οι περισσότερες από τις υπάρχουσες φορητές/κινητές συσκευές χρησιμοποιούν φθηνούς, κακώς βαθμονομημένους αισθητήρες, και, ως αποτέλεσμα, οι αναφορές από έναν αισθητήρα σπάνια συμβαδίζουν απόλυτα με τις αντίστοιχες αναφορές ενός άλλου για την ίδια μέτρηση. Συνεπώς, η ετερογένεια στις μετρήσεις μπορεί να οδηγήσει στην ανάπτυξη διαφορετικού μοντέλου Βαθιάς Μάθησης [166].



- Τα δεδομένα εισόδου στα Συνελικτικά Νευρωνικά Δίκτυα πρέπει να έχουν σταθερές διαστάσεις. Στο πρόβλημα της αναγνώρισης ανθρώπινης δραστηριότητας ο ετερογενής ρυθμός δειγματοληψίας, δηλαδή η μη χρήση της ίδιας συχνότητας δειγματοληψίας (διαφορετικά χρονικά παράθυρα), οδηγεί σε διαφορετικές αρχιτεκτονικές μοντέλων Βαθιάς Μάθησης. Για παράδειγμα, ένα CNN εκπαιδευμένο σε δείγματα που συλλέχθηκαν με συχνότητα δειγματοληψίας 200 Hz δεν μπορεί να χρησιμοποιηθεί για την ταξινόμηση δειγμάτων που συλλέγονται με συχνότητα δειγματοληψίας 50 Hz. Έτσι, θα πρέπει να υιοθετηθούν τεχνικές ομαδοποίησης (clustering), προκειμένου να οριστούν αρχιτεκτονικές CNN με δυνατότητα γενίκευσης.

### 7.3 Μελλοντική Έρευνα

Στην παρούσα ενότητα γίνεται λόγος για τα ανοικτά προς μελλοντική μελέτη θέματα, έχοντας ως βάση το προτεινόμενο SIoT οικοσυστήματος. Η συγκεκριμένη ανάλυση γίνεται με βάση τρεις γνώμονες: α) τη διαχείριση φιλίας, β) τη διαχείριση στόχων και γ) τις τεχνικές Τεχνητής Νοημοσύνης.

#### 7.3.1 Διαχείριση Φιλίας

Η διαχείριση της φιλίας έχει μεγάλη σημασία για την κοινωνική ζωή της έξυπνης οντότητας και, κατά επέκταση, για τον ιδιοκτήτη της. Μία έξυπνη οντότητα (συσκευή, πράκτορας λογισμικού ή εφαρμογή) δεν είναι δυνατόν να έχει σχέση φιλίας με άπειρο αριθμό άλλων έξυπνων οντοτήτων, διότι αυτό θα οδηγήσει σε ζητήματα κλιμακώσεως με επιπτώσεις την υπερφόρτωση μνήμης και τις καθυστερήσεις στο δίκτυο. Το *reasoner* είναι το συστατικό στοιχείο λογισμικού του SIoT Agent που γνωρίζει με ποιες έξυπνες οντότητες σχετίζεται ο τελευταίος, καθώς και το είδος της σχέσης που μοιράζονται. Παράλληλα, είναι σε θέση να αναλάβει δράση για τη δημιουργία φιλίας, τον τερματισμό ή την προσθήκη νέων σχέσεων. Για τη διαχείριση της φιλίας, ένα γνωσιακός πράκτορας πρέπει να λαμβάνει υπόψη όχι μόνο τη δομή του Δικτύου [34] (Ενότητα 2.1.1.2), αλλά και την Ποιότητα των Δεδομένων (Quality of Data; QoD) [40] του κόμβου, το βαθμό εμπιστοσύνης, την Ποιότητα των Πληροφοριών (Quality of Information; QoI), και τη φύση του κόμβου με βάση τη θεωρία παιγνίων. Συνεπώς, οι παραπάνω παράμετροι ορίζουν πόσο σημαντικός είναι ένας φίλος (*hasImportanceValue*) για μία έξυπνη οντότητα.

Η QoD μπορεί να υπολογιστεί μέσω εφαρμογών Συλλογικής Ανίχνευσης Δεδομένων (Crowdsensing). Οι μετρήσεις των αισθητήρων χωρίζονται σε δύο κατηγορίες: άμεσες μετρήσεις, στις οποίες τα εισερχόμενα δεδομένα προέρχονται από μία πραγματικά μετρούμενη παράμετρο και έμμεσες μετρήσεις, στις οποίες η παρατήρηση προέρχεται από συμπεράσματα υπό αβεβαιότητα σχετικά με την

πραγματική τιμή μίας παραμέτρου. Η συγκέντρωση δεδομένων είναι ένα δύσκολο ζήτημα, ακόμη και αν αυτά έχουν προηγουμένως μεταφραστεί σε χρήσιμα δεδομένα. Και οι δύο περιπτώσεις απαιτούν τεχνικές στατιστικής, που θα συγκεντρώνουν όλες τις μετρήσεις που αφορούν την ίδια μεταβλητή σε μία τιμή. Στις περισσότερες περιπτώσεις, αποκλειστικά και μόνο ο υπολογισμός του μέσου όρου αυτών των μετρήσεων δεν είναι η καλύτερη λύση. Για παράδειγμα, έστω ότι μία εφαρμογή έξυπνου τηλεφώνου ισχυρίζεται ότι ο κάτοχός της έχει περπατήσει μία απόσταση 3 χιλιομέτρων, ενώ μία φορητή συσκευή (wearable device) έχει εντοπίσει 5 χιλιόμετρα και μία άλλη 4,8 χιλιόμετρα για την ίδια ημέρα. Σε αυτό το παράδειγμα, τουλάχιστον μία παρατήρηση είναι θορυβώδης. Ένας πράκτορας γυμναστικής που είναι φίλος τους πρέπει να συγκεντρώσει τις αξίες τους. Υπάρχουν πολλοί τρόποι με τους οποίους ο πράκτορας μπορεί να ρυθμίσει τη συνολική τιμή της μετρούμενης απόστασης. Μία ευρετική προσέγγιση είναι να δοθεί μεγαλύτερο βάρος στις φορητές συσκευές, επειδή ο χρήστης δεν μεταφέρει μαζί του όλο το χρόνο το έξυπνο τηλέφωνο. Μία άλλη προσέγγιση είναι να προτιμηθεί η βασισμένη στην πλειοψηφία απόφαση, σύμφωνα με την οποία ο πράκτορας θα λάβει υπόψη μόνο την αξία των υψηλότερων τιμών, συν ή πλην μίας τιμής σφάλματος. Στο παραπάνω παράδειγμα, εάν η τιμή σφάλματος είναι 0,05 για κάθε χιλιόμετρο, τότε το σφάλμα για 5 χλμ. είναι 0,25 χλμ. Συνεπώς, οι δύο φορητές συσκευές θεωρούνται ότι συγκλίνουν. Μία τελευταία προσέγγιση που είναι κατάλληλη για την αξιολόγηση των δεδομένων του IoT είναι ο προσδιορισμός των θορυβωδών μετρήσεων κάνοντας χρήση της Επιβλεπόμενης Μάθησης (Supervised Learning). Μέσω της Ποιότητας της Εμπειρίας (Quality of Experience; QoE), ο πράκτορας λογισμικού θα αρχίσει να σταθμίζει τις μετρήσεις των συσκευών. Για παράδειγμα, ένας πράκτορας λογισμικού για τον καιρό έχει δύο φίλους για την πρόβλεψη βροχής: μία εφαρμογή εγκατεστημένη στο έξυπνο τηλέφωνο του ιδιοκτήτη του και μία εφαρμογή εγκατεστημένη σε ένα έξυπνο αυτοκίνητο που μοιράζονται μία σχέση συνεργασίας. Μερικές φορές, οι πιθανότητες βροχής κατά τη διάρκεια της ημέρας και ανάλογα με τη χρονική στιγμή, κατά την οποία αναφέρονται στον πράκτορα, αποκλίνουν. Με την ανάκτηση πραγματικών δεδομένων βροχής από έναν αισθητήρα υγρασίας, ο πράκτορας θα είναι σε θέση μετά από μία μικρή χρονική περίοδο να προσθέσει ένα βάρος στις δύο πιθανότητες βροχής. Μεγαλύτερο βάρος θα σημαίνει και μεγαλύτερη αξιοπιστία στη μέτρηση/πιθανότητα.

Η εμπιστοσύνη είναι ένας ακόμη σημαντικός παράγοντας για την επιλογή φίλων. Ακόμη και στο οικοσύστημα του IoT, ένας κόμβος μπορεί να είναι κακόβουλος. Στο [40] οι συγγραφείς προτείνουν μία τεχνική Ανίχνευσης Περιπτώσεων Άτυπης Συμπεριφοράς (Outlier Analysis) για την ταξινόμηση πιθανών απειλών σχετικά με την ασφάλεια και την ιδιωτικότητα. Για παράδειγμα, ένα όχημα μπορεί να δώσει ψευδή δεδομένα σχετικά με την κυκλοφορία η οποία σχετίζεται με τη διαδρομή που πρόκειται να πάρει, σε μία προσπάθειά του να κάνει τα άλλα οχήματα να επιλέξουν

άλλους δρόμους και ως εκ τούτου το ταξίδι του να είναι πιο σύντομο. Η Ανίχνευση Περιπτώσεων Άτυπης Συμπεριφοράς ή Ανίχνευσης Ανωμαλιών (Anomaly Detection) είναι, συνήθως, μία μη-επιβλεπόμενη μέθοδος Μηχανικής Μάθησης, στην οποία τα σημεία που έχουν μεγάλη απόσταση από το σύμπλεγμα, το οποίο σχηματίζουν τα υπόλοιπα, θεωρούνται απομακρυσμένα. Πιο συγκεκριμένα, η Ανίχνευση Ανωμαλίας μοντελοποιεί την πιθανότητα ενός συνόλου δειγμάτων εκπαίδευσης  $p(x)$  που θεωρούνται αξιόπιστα. Αν για ένα νέο δείγμα  $x_{test}$  ισχύει ότι το  $p(x_{test}) < e$ , όπου  $e$  είναι ένα σωστά ορισμένο όριο (συνήθως είναι πολύ μικρό νούμερο), τότε το δείγμα  $x$  επισημαίνεται ως ανωμαλία. Οι αποκλίνουσες τιμές χαρακτηρίζονται σημασιολογικά ως έχουσες κακή φήμη.

Τα τελευταία χρόνια, έχουν εφαρμοστεί στα δίκτυα προσεγγίσεις που βασίζονται στη θεωρία παιγνίων και έχουν αποδειχθεί αποτελεσματικές [159]. Το IoT είναι ένα καταναμημένο δυναμικό οικοσύστημα στο οποίο τα αντικείμενα μπορούν να θεωρηθούν ως οι παίκτες ενός παιχνιδιού. Ένα παιχνίδι αποτελείται από ένα πεπερασμένο σύνολο παικτών  $N = \{1, 2, \dots, n\}$  και ένα σύνολο ενεργειών  $A = \{1, 2, \dots, k\}$ , όπου κάθε παίκτης επιλέγει μία ενέργεια  $a \in A$  σε μία προσπάθειά του να μεγιστοποιήσει τη συνάρτηση ωφέλειας του  $u$ . Σύμφωνα με το SIoT, οι κόμβοι καταναλώνουν υπηρεσίες που προσφέρουν άλλοι κόμβοι. Ωστόσο, μερικές φορές κάποια αντικείμενα τείνουν να έχουν μία αποκαλούμενη εγωιστική συμπεριφορά. Λόγω προβλημάτων μπαταρίας, οι ιδιοκτήτες τους έχουν εισαγάγει περιορισμούς ισχύος ή απλά επειδή δεν δημοσιεύουν αξιόλογες υπηρεσίες, τείνουν να καταναλώνουν πολύ περισσότερες από όσες παράγουν. Στη θεωρία των παιχνιδιών, οι παίκτες θεωρούνται ορθολογικοί, οπότε ένα ορθολογικό αντικείμενο δεν πρέπει να έχει γειτονικούς κόμβους οι οποίοι δεν προσφέρουν αξιόπιστες υπηρεσίες ή καθόλου υπηρεσίες. Η ωφέλεια  $u_i \in [0, 1]$  μίας κοινωνικής σχέσης που διατηρεί ένα αντικείμενο θα πρέπει να υπολογίζεται σύμφωνα με την Ιδιότητα Δεδομένων  $wasActiveAt$ . Τα αντικείμενα που είναι ανενεργά για μεγάλο χρονικό διάστημα θα πρέπει να έχουν χαμηλή ωφέλεια και τα αντικείμενα που τείνουν να συνεργάζονται (δηλ., δεν είναι εγωιστικά σε όρους θεωρίας παιγνίων) θα πρέπει να έχουν υψηλή ωφέλεια. Η ενέργεια που έχει οριστεί σε αυτήν την προσέγγιση έχει δύο τιμές όσον αφορά τα δεδομένα μίας υπηρεσίας  $\{\text{διαμοιράζω}, \text{δεν διαμοιράζω}\}$ . Σε ορισμένες περιπτώσεις, ένας κόμβος μπορεί να μην προσφέρει πολύτιμες υπηρεσίες, αλλά μπορεί να έχει πολλούς φίλους (κεντρικότητα) ή να είναι ένας διαδότης πληροφοριών που έχει μεγάλη επιρροή (influential spreader). Ως αποτέλεσμα, ακόμη και αν ένα αντικείμενο δεν καταναλώνει τις υπηρεσίες ενός άλλου, η σχέση τους θεωρείται ενεργή αν το συγκεκριμένο αντικείμενο παίζει το ρόλο του διαμεσολαβητή, συνδέοντας το με τους φίλους του.

Η QoI είναι ένα άλλο χαρακτηριστικό που πρέπει να ληφθεί υπόψη για τη σημασία της φιλίας [40][160]. Σε αντίθεση με την QoD που παρουσιάζει την ακρίβεια των δεδομένων, η QoI υποδηλώνει την αξία των πληροφοριών που παρέχει ένα

αντικείμενο ή ένας πράκτορας λογισμικού. Πιο συγκεκριμένα, απεικονίζει πόσο σημαντικές είναι οι πληροφορίες που λαμβάνει η έξυπνη οντότητα από τους γείτονές της προκειμένου να πάρει μία απόφαση. Μία προτεινόμενη τιμή [160] για τη  $QoI$  δίνεται από την:

$$QoI = Q * P * R * A * D * T * V \quad (7.1)$$

όπου το  $Q$  υποδηλώνει την ποσότητα της πληροφορίας, το  $P$  υποδηλώνει την ακρίβεια, το  $R$  υποδηλώνει την ανάκληση, το  $A$  υποδηλώνει την ορθότητα, το  $D$  υποδηλώνει λεπτομέρεια, το  $T$  υποδηλώνει την εγκαιρότητα και το  $V$  υποδηλώνει την εγκυρότητα. Πρέπει να αναφερθεί ότι όλες οι τιμές κανονικοποιούνται στο  $[0, 1]$ . Στην (7.1) η ποσότητα αντιπροσωπεύει πόση χρήσιμη πληροφορία έχει λάβει ο υπεύθυνος λήψης αποφάσεων για μία συγκεκριμένη εργασία. Για παράδειγμα, για έναν πράκτορα υγείας, ο καρδιακός ρυθμός που λαμβάνει από μία φορητή συσκευή είναι πολύτιμος. Η ακρίβεια υποδηλώνει την αναλογία του αριθμού των σχετικών πληροφοριών που ανακτώνται προς το συνολικό αριθμό άσχετων και σχετικών πληροφοριών που ανακτώνται, ενώ η ανάκληση είναι ο λόγος του αριθμού των σχετικών πληροφοριών που ανακτώνται προς το συνολικό αριθμό των σχετικών πληροφοριών. Η ορθότητα απεικονίζει το βαθμό αξιοπιστίας των πληροφοριών και η λεπτομέρεια αναπαριστά το συνολικό βαθμό των πληροφοριών που αποστέλλονται στον υπεύθυνο λήψης αποφάσεων. Η εγκαιρότητα είναι το αντίθετο της καθυστέρησης. Ως εκ τούτου, μία τιμή ίση με 1 σημαίνει ότι οι πληροφορίες, που χρειάζεται ο υπεύθυνος λήψης αποφάσεων, ήρθαν εγκαίρως. Δηλαδή, αν οι πληροφορίες είναι διαθέσιμες πριν από τη λήψη απόφασης από τον υπεύθυνο λήψης αποφάσεων, η εγκαιρότητα ισούται με 1. Τέλος, η εγκυρότητα ισούται με 1, εάν η παρεχόμενη πληροφορία είναι αληθής και με μηδέν, εάν είναι εντελώς ψευδής.

### 7.3.2 Διαχείριση Στόχων

#### 7.3.2.1 Ανακάλυψη Υπηρεσίας

Ο μηχανισμός *Ανακάλυψης Υπηρεσίας* είναι ένα πολύ σημαντικό συστατικό στοιχείο λογισμικού, το οποίο βοηθάει την έξυπνη οντότητα πρώτα να αναζητήσει και στη συνέχεια να εγγραφεί σε υπηρεσίες που προσφέρουν άλλες έξυπνες οντότητες. Ο μηχανισμός *Ανακάλυψης Υπηρεσίας* είναι υπεύθυνος για την εύρεση των θεμάτων στην πλατφόρμα του *SlOT* τα οποία καλύπτουν τις ανάγκες της οντότητας που αυτό εκπροσωπεί. Αρχικά, ο έξυπνος πράκτορας λογισμικού ελέγχει τη λίστα των φίλων του. Εάν κανένας από τους φίλους του δεν προσφέρει το είδος της πληροφορίας που αναζητά, ο πράκτορας ξεκινά τη διαδικασία *Ανακάλυψης Υπηρεσίας*. Η διαδικασία αυτή πραγματοποιείται στην παρούσα διατριβή με χρήση σημασιολογικών ερωτημάτων *SPARQL*, μέσω των οποίων ο πράκτορας λογισμικού ζητά τις υπηρεσίες των φίλων των φίλων του. Παρόλα αυτά, για την *Ανακάλυψη*

Υπηρεσίας μπορούν να χρησιμοποιηθούν και άλλες τεχνικές αναζήτησης με βάση: α) την τοποθεσία και β) το ίδιο Δίκτυο (Ad-hoc Δίκτυο ή MANET).

Η τεχνική *Ανακάλυψης Υπηρεσίας* που βασίζεται σε γεωγραφικές περιοχές είναι σημαντική για το «κλάδεμα» του χώρου αναζήτησης, ειδικά σε περιπτώσεις όπου ο πράκτορας, που μπορεί να ικανοποιήσει την ανάγκη του αιτούντος, βρίσκεται σε απομακρυσμένη θέση. Για παράδειγμα, ο John ζει στο Σαν Φρανσίσκο και θα επισκεφθεί το Παρίσι σε λίγες ημέρες, χωρίς να διαθέτει τον απαραίτητο χρόνο για να προγραμματίσει το ταξίδι του. Ως εκ τούτου, αγόρασε μία προηγμένη εφαρμογή που λειτουργεί σαν ταξιδιωτικός πράκτορας και την κατέγραψε κάτω από το προφίλ του στην πλατφόρμα *SlOT*. Αυτόματα ο ταξιδιωτικός πράκτορας κάνει αιτήματα φιλίας προς τα άλλα αντικείμενα ή εφαρμογές που κατέχει ο John. Μετά από αυτό, καθιερώνει μία Σχέση Ιδιοκτησίας Αντικειμένου με τα περισσότερα από αυτά. Ο ταξιδιωτικός πράκτορας θέλει να βρει την πιο κατάλληλη ημέρα για να επισκεφθεί ο John τον Πύργο του Άιφελ. Για αυτό ζητά από τον πράκτορα καιρού (που έχει προσθέσει ως φίλο), να τον ενημερώσει σχετικά με την εικόνα του καιρού στο Παρίσι κατά τη διάρκεια της παραμονής του John και τότε ο πράκτορας καιρού του επιστρέφει μία εκτίμηση για τις καιρικές συνθήκες που θα επικρατήσουν. Εκτός από αυτό, ο ταξιδιωτικός πράκτορας θέλει να βρει την ημέρα και ώρα που ο Πύργος του Άιφελ έχει το χαμηλότερο αριθμό επισκεπτών. Δυστυχώς, κανένας από τους φίλους του δεν διαθέτει τέτοιες πληροφορίες, για αυτό τους ρωτάει αν έχουν έναν φίλο που παρέχει αυτό το είδος πληροφοριών. Φυσικά είναι πολύ δύσκολο να ζητήσει απευθείας αυτές τις πληροφορίες. Ένας τρόπος επίλυσης είναι η χρήση Πρακτόρων Διαμεσολάβησης που καλύπτουν ένα σύμπλεγμα οντοτήτων, οι οποίες βρίσκονται σε συγκεκριμένο γεωγραφικό πλάτος και μήκος (π.χ., έχουν ως τοποθεσία το Παρίσι), καθώς θα ήταν αδύνατον μία και μόνο πλατφόρμα να καλύψει την επικοινωνία δισεκατομμυρίων οντοτήτων του *IoE*. Με αυτόν τον τρόπο ο ταξιδιωτικός πράκτορας ζητά από τον Πράκτορα Διαμεσολάβησης του συμπλέγματος στο οποίο βρίσκεται (Σαν Φρανσίσκο), να του επιστρέψει την ηλεκτρονική διεύθυνση του Πράκτορα Διαμεσολάβησης που είναι υπεύθυνος για την κάλυψη των περιοχών κοντά στο γεωγραφικό πλάτος και το γεωγραφικό μήκος του Πύργου του Άιφελ. Στη συνέχεια, ο ταξιδιωτικός πράκτορας αποστέλλει στον Πράκτορα Διαμεσολάβησης του Παρισινού συμπλέγματος ένα αίτημα επιστροφής των ηλεκτρονικών διευθύνσεων των αντικειμένων κοντά στον Πύργο του Άιφελ. Ένας από αυτούς είναι ένας αισθητήρας παρουσίας που βρίσκεται έξω από τον Πύργο. Τέλος, ο ταξιδιωτικός πράκτορας προωθεί ένα αίτημα φιλίας στον πράκτορα του αισθητήρα, πραγματοποιεί συνδρομή στις υπηρεσίες του και κατεβάζει τα δεδομένα που σχετίζονται με τον αριθμό των επισκεπτών του Πύργου του Άιφελ. Μέσω μηχανισμών βελτιστοποίησης, ο ταξιδιωτικός πράκτορας επιλέγει την πιο κατάλληλη ημέρα και ώρα για να επισκεφθεί ο John το παριζιάνικο μνημείο.

Η *Ανακάλυψη Υπηρεσίας* στα MANET έχει μελετηθεί αρκετές φορές [162][163][164]. Η δυναμική και κατανεμημένη φύση των MANET επιτρέπει τη χρήση πολλών τεχνικών, που κυμαίνονται από τις παραδοσιακές Δρομολογήσεις δια Πλημμύρας (Flooding) [162] έως τους μεμονωμένους Τυχαίους Περιπατητές (Random Walkers) [163]. Η προηγούμενη λύση καταφέρνει να καλύψει το δίκτυο, αλλά έχει μικρή επεκτασιμότητα λόγω του μεγάλου αριθμού των πακέτων που αυξάνεται μαζί με το μέγεθος του δικτύου, ενώ η τελευταία λύση αντιμετωπίζει αποτελεσματικότερα αυτό το πρόβλημα των παραγόμενων πακέτων, αλλά επηρεάζει ιδιαίτερα το χρόνο που απαιτείται για την κάλυψη του συνολικού δικτύου. Προκειμένου να καλυφθεί το κενό μεταξύ αυτών των δύο προσεγγίσεων, χρησιμοποιούνται συγκεκριμένες τεχνικές, όπως αυτές των πιθανοτικών Δρομολογήσεων δια Πλημμύρας και των πολλαπλών Τυχαίων Περιπατητών. Η επιλογή της βέλτιστης τεχνικής δεν είναι εύκολη και μπορεί να εξαρτάται όχι μόνο από τον αριθμό των πακέτων και το χρόνο που απαιτείται για την κάλυψη του δικτύου, αλλά και από άλλους παράγοντες, όπως η κατανάλωση ενέργειας και η δυναμική τοπολογία του δικτύου [164]. Στο οικοσύστημα του *SlOT*, μία έξυπνη συσκευή μπορεί να συναντήσει περιστασιακούς φίλους (*Occasional Friends*), με τους οποίους μπορεί να δημιουργήσει ένα MANET ή VANET, και μπορεί να τους χρησιμοποιήσει για να μάθει ποιοι άλλοι μπορούν να καλύψουν τις ανάγκες της, όπως, για παράδειγμα, οι πράκτορες λογισμικού που μπορούν να παρέχουν ενημερωμένες πληροφορίες σχετικά με την κίνηση σε έναν δρόμο, όπου έχει συμβεί κάποιο ατύχημα λίγα λεπτά πριν.

### 7.3.2.2 *Λήψη Αποφάσεων*

Ο μηχανισμός *Λήψης Αποφάσεων* είναι, επίσης, πολύ σημαντικός και για αυτό το λόγο επιλέχθηκε η χρήση παραγωγικού συλλογισμού πρώτου βαθμού σε οντολογικά χαρακτηρισμένα δεδομένα. Έτσι, ο τρόπος που σκέφτονται και παίρνουν αποφάσεις οι μηχανές μοιάζει με αυτόν των ανθρώπων. Ένας άλλος δημοφιλής μηχανισμός που θα μπορούσαν να χρησιμοποιήσουν οι πράκτορες λογισμικού για τη λήψη αποφάσεων είναι η Ενισχυτική Μάθηση (Reinforcement Learning) [165]. Τα μοντέλα Ενισχυτικής Μάθησης επιλέγουν τη βέλτιστη πολιτική που πρέπει να ακολουθήσουν, μεταβαίνοντας από την κατάσταση στην οποία βρίσκονται σε μία άλλη, στην προσπάθειά τους να μεγιστοποιήσουν την ωφέλεια που θα λάβει ο πράκτορας. Επίσης, είναι κατάλληλα για περιπτώσεις όπου το μοντέλο μετάβασης και ανταμοιβής δεν είναι διαθέσιμο (μέθοδος Q-learning).

Όμως μέχρι το Μάρτιο του 2016 [166], όπου το πρόγραμμα λογισμικού AlphaGo κέρδισε τον επί χρόνια πρωταθλητή στο παιχνίδι Go, οι αλγόριθμοι Ενισχυτικής Μάθησης έδειχναν ακατάλληλοι για περίπλοκα οικοσυστήματα, όπως αυτό του *SlOT*. Συγκεκριμένα, το AlphaGo βασίζεται στον αλγόριθμο Κλίσης Πολιτικής (Policy Gradient) [167], μία τεχνική Βαθιάς Ενισχυτικής Μάθησης (Deep Reinforcement Learning). Δεδομένου ότι οι τεχνικές Βαθιάς Μάθησης επιτρέπουν στους πράκτορες

λογισμικού να μιλούν και να κατανοούν την ανθρώπινη γλώσσα [168], οι SIoT Agents θα μπορούσαν να κάνουν χρήση του αλγορίθμου Policy Gradient στην προσπάθειά τους να επικοινωνήσουν μεταξύ τους και με τους ανθρώπους με τέτοιο τρόπο, ώστε να μεγιστοποιήσουν τη συνάρτηση ωφέλειάς τους, πετυχαίνοντας τους στόχους τους [169][170].

### 7.3.3 Τεχνικές Τεχνητής Νοημοσύνης

Παρά τα πολλά υποσχόμενα αποτελέσματα της χρήσης μοντέλων Βαθιάς Μάθησης για την εξαγωγή πληροφοριών πλαισίου, βασισμένων σε αισθητήρες κίνησης φορητών/κινητών συσκευών, υπάρχουν μηχανισμοί που δεν μελετήθηκαν στην παρούσα διατριβή (Πίνακας 17), όπως οι εξής: α) Απομνημόνευση, και β) Ημι-επιβλεπόμενη Μάθηση. Οι συγκεκριμένοι μηχανισμοί, αν και δεν έχουν βρει εφαρμογή στο πεδίο της αναγνώρισης ανθρώπινων δραστηριοτήτων, έχουν εφαρμοστεί με επιτυχία σε άλλα υπολογιστικά προβλήματα (π.χ., υπολογιστική όραση), και είναι πολλά υποσχόμενοι όσον αφορά τη μελλοντική έρευνα στο συγκεκριμένο πεδίο.

Η χρήση μηχανισμών απομνημόνευσης σε αλγόριθμους Βαθιάς Μάθησης αποτελεί ένα ερευνητικό πεδίο το οποίο προσελκύει το ενδιαφέρον πολλών ερευνητών τα τελευταία χρόνια. Επιπλέον κίνητρο προς αυτή την κατεύθυνση δίνεται από το γεγονός ότι, για να εκπαιδευτεί ένας αλγόριθμος ταξινόμησης που βασίζεται σε Βαθιά Μάθηση, απαιτείται η συλλογή και η επεξεργασία χιλιάδων δειγμάτων για κάθε κλάση/περίπτωση. Ο μηχανισμός Μηχανικής Μάθησης βάσει του οποίου ένα μοντέλο μπορεί να προβλέψει την κλάση ενός δείγματος έχοντας εκπαιδευτεί σε ελάχιστα όμοια με αυτό δείγματα (π.χ., 1-10), ονομάζεται Μάθηση με μία Προσπάθεια (One-Shot Learning). Οι τεχνικές One-Shot Learning, όπως τα Σιαμαία Νευρωνικά Δίκτυα (Siamese Neural Networks) [171], έχουν εφαρμοστεί κατά κόρον σε προβλήματα υπολογιστικής όρασης. Μία αξιόλογη ερευνητική δουλειά αποτελεί ο Διαφοροποιήσιμος Νευρωνικός Υπολογιστής (Differentiable Neural Computer) [172], ο οποίος όχι μόνο κάνει χρήση μηχανισμού απομνημόνευσης, αλλά μπορεί να εφαρμοστεί και σε πολλά διαφορετικά υπολογιστικά προβλήματα (π.χ., συλλογισμός σε οντολογικά δεδομένα) χωρίς να αλλάζει η αρχιτεκτονική του Νευρωνικού Δικτύου.

Το μειονέκτημα της Επιβλεπόμενης Μάθησης είναι η απαίτηση ύπαρξης ετικέτας (τίτλου) σε όλα τα δείγματα κατά την εκπαίδευση του μοντέλου. Για αυτό το λόγο, τεχνικές Ημι-Επιβλεπόμενης Μάθησης (Semi-Supervised Learning) έχουν προταθεί προσφάτως για την εκπαίδευση μοντέλων Βαθιάς Μάθησης. Παράδειγμα τέτοιας μάθησης, όσον αφορά το πρόβλημα της αναγνώρισης ήχου, αποτελεί η ερευνητική εργασία [173]. Οι συγγραφείς προτείνουν την τιτλοδότηση των ηχητικών σημάτων κάνοντας χρήση βίντεο. Κατά την εκπαίδευση του μοντέλου, τα μισά συλλεχθέντα ηχητικά σήματα δεν ήταν συγχρονισμένα με το βίντεο (π.χ., υπήρχε

χρονοκαθυστέρηση στον ήχο ή την εικόνα). Ως αποτέλεσμα, η εκπαίδευση του μοντέλου είχε, επιπλέον, ως στόχο το διαχωρισμό των συγχρονισμένων από τα ασυγχρόνιστα δείγματα. Χρήση παρόμοιας τεχνικής βασισμένης στη συγχώνευση βίντεο και σημάτων κίνησης θα μπορούσε να μελετηθεί και σε προβλήματα αναγνώρισης ανθρώπινων δραστηριοτήτων.



## 8 Αναφορές

- [1] D. D. Guinard and V. M. Trifa, “Building the Web of Things,” Manning, June 2016.
- [2] Carnot Institutes, “Smart networked objects and internet of things,” Carnot Institutes’ Information Communication Technologies and Micro Nano Technologies alliance, White Paper, January 2011, [http://www.internet-of-things-research.eu/pdf/IoT\\_Clusterboo\\_March\\_2010.pdf](http://www.internet-of-things-research.eu/pdf/IoT_Clusterboo_March_2010.pdf). Last accessed at 20/11/2017.
- [3] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, “Smart objects as building blocks for the internet of things,” *IEEE Internet Computing*, vol. 14, no. 1, pp. 44–51, 2010.
- [4] D. Le-Phuoc, A. Polleres, M. Hauswirth, G. Tummarello, and C. Morbidoni, “Rapid prototyping of semantic mash-ups through semantic web pipes,” in *Proc. 18th international conference on World wide web*, ser. WWW 2009. ACM, pp. 581–590, 2009.
- [5] A. Dohr, R. Modre-Opsrian, M. Drobics, D. Hayn, and G. Schreier, “The internet of things for ambient assisted living,” in *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, pp. 804–809, 2010.
- [6] Bringing Things to Life with IoT. Available at: <https://blog.intuz.com/bringing-things-to-life-with-iot/>. Last accessed at 26/10/2017.
- [7] A. Zaslavsky, C. Perera, and D. Georgakopoulos, “Sensing as a service and big data,” in *International Conference on Advances in Cloud Computing (ACC-2012)*, Bangalore, India, pp. 21–29, July 2012.
- [8] On line dictionary, “Socialization” lemma, available at URL: [dictionary.reference.com/](http://dictionary.reference.com/). Last accessed at 27/03/2016.
- [9] L. Atzori, A. Iera, and G. Morabito, “SIoT: Giving a social structure to the internet of things,” *Communications Letters, IEEE*, Vol. 15, No. 11, pp. 1193–1195, Nov. 2011.
- [10] Y. LeCun, Y. Bengio & G. Hinton, “Deep Learning”, in *Nature* vol. 521, pp. 436–444, May 2015.
- [11] A. Krizhevsky , I. Sutskever , G. E. Hinton, “Imagenet Classification with Deep Convolutional Neural Networks”, *Advances in Neural Information Processing Systems*, 2012.
- [12] A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Proceedings of ICASSP 2013*, Vancouver, Canada, May 2013.
- [13] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng and C. Potts, “Recursive deep models for semantic compositionality over a sentiment

- treebank,” in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631-1642, 2013.
- [14] P. Kasnesis, C. Z. Patrikakis, I. S. Venieris, “Changing the Game of Mobile Data Analysis with Deep Learning,” IEEE IPro Magazine., vol. 19, pp. 17–23, 2017.
- [15] P. Vincent, H. Larochelle Y. Bengio and P.A. Manzagol, “Extracting and Composing Robust Features with Denoising Autoencoders,” in Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML’08), pp. 1096-1103, ACM, 2008.
- [16] S. Bosems, M. Sinderen, “Model-driven development for user-centric well-being support from dynamic well-being domain models to context-aware applications,” in 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD), pp. 425-432, 2015.
- [17] F. Ongenaes, M. Claeys, T. Dupont, W. Kerckhove, P. Verhoeve, T. Dhaene, F. D. Turck, “A probabilistic ontology-based platform for self-learning context-aware healthcare applications,” in Expert Systems with Applications, vol. 40, no. 18, pp. 7629-7646, December 2013.
- [18] W. Li, A. Joshi, T. Finin, “SVM-CASE: An SVM-Based Context Aware Security Framework for Vehicular Ad-Hoc Networks,” in IEEE 82nd Vehicular Technology Conference (VTC2015-Fall), pp. 1-5, 2015.
- [19] P. Kasnesis, L. Toumanidis, D. Kogias, C. Patrikakis and I. S. Venieris, “ASSIST: An Agent-Based SIoT Simulator,” in the Proceedings of the 2016 IEEE World Forum on Internet of Things, pp. 353-358, 2016.
- [20] N. Olifer and V. Olifer, “Computer Networks: Principles, Technologies and Protocols for Network Design,” in John Wiley & Sons, 2005.
- [21] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, Member, “Context Aware Computing for The Internet of Things: A Survey, ” in IEEE Communications Surveys & Tutorials, vol. 16, no. 1 pp. 414-454, 2014.
- [22] K. Ashton, “That ‘internet of things’ thing in the real world, things matter more than ideas,” RFID Journal, June 2009, <http://www.rfidjournal.com/article/print/4986>. Last accessed at 20/11/2017.
- [23] International Telecommunication Union, “Itu internet reports 2005: The internet of things,” International Telecommunication Union, Workshop Report, November 2005, <http://www.itu.int/dms pub/itu-s/opb/pol/SPOL-IR.IT-2005-SUM-PDF-E.pdf>. Last accessed at 20/11/2017.
- [24] T. Lu and W. Neng, “Future internet: The internet of things,” in 3rd International Conference on Advanced Computer Theory and Engineering(ICAETE), vol. 5, pp. 376-380, August 2010.
- [25] P. Guillemin and P. Friess, “Internet of things strategic research roadmap,” The Cluster of European Research Projects, Tech. Rep., September 2009, [http://www.internet-of-things-research.eu/pdf/IoT\\_Cluster\\_Strategic\\_Research\\_Agenda\\_2009.pdf](http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2009.pdf). Last accessed at 20/11/2017.

- [26] P. Banerjee, R. Friedrich, C. Bash, P. Goldsack, B. Huberman, J. Manley, C. Patel, P. Ranganathan, and A. Veitch, "Everything as a service: Powering the new information economy," *IEEE Computer*, vol. 44, no. 3, pp. 36–43, 2011.
- [27] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kroller, M. Pagel, M. Hauswirth, M. Karnstedt, M. Leggieri, A. Passant, R. Richardson, "Spitfire: toward a semantic web of things," in *Communications Magazine, IEEE*, Vol. 49, pp. 40–48, 2011.
- [28] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web. *Scientific American*," vol. 284, no. 5, pp. 34-43, 2001.
- [29] RDFa. Available at: <https://rdfa.info/>. Last accessed at 22/08/2017.
- [30] JSON-LD. Available at: <https://json-ld.org/>. Last accessed at 22/08/2017.
- [31] Hypercat specification. Available at: <http://www.hypercat.io/>. Last accessed at 22/08/2017.
- [32] J. Travers and S. Milgram. "An Experimental Study of the Small World Problem." *Sociometry*, Vol. 32, No. 4, pp. 425-443, 1969.
- [33] L. Atzori, A. Iera, G. Morabito, M. Nitti, The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization, *Elsevier Computer Networks*, Vol. 56, Issue 16, pp. 3594–3608, 2012.
- [34] M. Nitti, L. Atzori, I. P. Cvijikj, Friendship Selection in the Social Internet of Things: Challenges and Possible Strategies, *IEEE Internet of Things Journal*, Vol. 2, No. 3, pp. 240-247, June 2015.
- [35] V. Beltran, A. M. Ortiz, D. Hussein, N. Crespi, "A Semantic Service Creation Platform for Social IoT", *Proceedings of IEEE World Forum on Internet of Things (WF-IoT)*, pp. 283–286, 2014.
- [36] D. Guinard, V. Trifa, E. Wilde, "A resource oriented architecture for the Web of Things", *Proceedings of Internet of Things (IoT)*, pp. 1-8, 2010.
- [37] J. E. Kim, A. Maron, D. Mosse, "Socialite: A Flexible Framework for Social Internet of Things", *Proceedings of 16th IEEE International Conference on Mobile Data Management*, pp. 94-103, 2015.
- [38] D. Hussein, S. N. Han, Gyu M. Lee, Noel Crespi, "Social Cloud-based Cognitive Reasoning for Task-oriented Recommendation in the Social Internet of Things," in *IEEE Cloud Computing*, vol. 2, no. 6, pp. 10-19, 2015.
- [39] M. Zhang, H. Zhao, R. Zheng, Q. Wu and W. Wei, "Cognitive Internet of Things: Concepts and Application Example," *International Journal of Computer Science Issues*, vol. 9, no. 6, pp. 151-158, 2012.
- [40] Q. Wu, G. Ding, Y. Xu, S. Feng, Z. Du, J. Wang, K. Long, "Cognitive Internet of Things: A New Paradigm Beyond Connection", *IEEE Internet of Things Journal*, Vol. 1, No. 2, pp. 129-143, 2014.
- [41] IBM Watson. Available at: <https://www.ibm.com/watson/>. Last accessed at 29/08/2017.

- [42] IBM Bluemix. Available at: <https://www.ibm.com/cloud-computing/bluemix/>. Last accessed at 29/08/2017.
- [43] IBM Cognitive IoT Architecture. Available at: <https://www.ibm.com/devops/method/content/architecture/cognitiveArchitecture/>. Last accessed at 29/08/2017.
- [44] P. Kasnesis, C. Patrikakis, D. Kogias, L. Toumanidis and I. S. Venieris, "Cognitive Friendship and Goal Management for the Social IoT," in the Computers & Electrical Engineering, Elsevier, Vol. 58, pp. 412–428, 2017.
- [45] R. Baldoni, R. M. Contenti, A. Virgillito, "The Evolution of Publish/Subscribe Communication Systems," Future Directions of Distributed Computing. Springer Verlag LNCS Vol. 2584, pp. 137-141, 2003.
- [46] MQTT message protocol. Available at: <http://mqtt.org/>. Last accessed at 02/09/2017.
- [47] World Wide Web Consortium (W3C), "Resource Description Framework (RDF)". Available at: <https://www.w3.org/RDF/>. Last accessed 21/11/2017
- [48] World Wide Web Consortium (W3C), "RDF Schema 1.1.". Available at: <https://www.w3.org/TR/rdf-schema>. Last accessed 21/11/2017
- [49] World Wide Web Consortium (W3C), "OWL Web Ontology Language Overview.". Available at: <http://www.w3.org/TR/2004/REC-owl-features-20040210/>. Last accessed 21/11/2017.
- [50] World Wide Web Consortium (W3C), "SPARQL 1.1 Query Language.". Available at: <http://www.w3.org/TR/sparql11-query>, Last accessed 21/11/2017.
- [51] World Wide Web Consortium (W3C), "OWL 2 Web Ontology Language Document Overview (Second Edition)". <http://www.w3.org/TR/owl2-overview/>, accessed March 2017.
- [52] T. Berners-Lee, "Linked Data," W3C, 2006.
- [53] D. Bonino, E. Castellina, F. Corno, "DOG: An Ontology-Powered OSGi Domestic Gateway", Tools with Artificial Intelligence, pp. 157-160, 2008.
- [54] D. Bonino, F. Corno, "DogOnt - Ontology Modeling for Intelligent Domestic Environments", The Semantic Web - ISWC 2008, pp 790-803, 2008.
- [55] L. Sommaruga, A. Perri, Fr. Furfari, "DomoML-env: an ontology for Human Home Interaction", Proceedings of SWAP, Vol-166, pp. 249-256, 2005.
- [56] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, et al., "The SSN Ontology of the W3C Semantic Sensor Network Incubator Group," in Web Semantics: Science, Services and Agents on the World Wide Web, vol. 17, pp. 25-32, 2012.
- [57] H. van der Schaaf, R. Herzog, "Mapping the OGC Sensor Things API onto the OpenIoT Middleware," in Interoperability and Open-Source Solutions for the Internet of Things: International Workshop, FP7 OpenIoT Project, Held in

- Conjunction with SoftCOM 2014, Springer International Publishing, Croatia, pp. 62–70, 2014.
- [58] R. Agarwal, D. G. Fernandez, T. Elsaleh, A. Gyrard, J. Lanza, L. Sanchez, N. Georgantas, V. Issarny, “Unified IoT Ontology to Enable Interoperability and Federation of Testbeds, in: 3rd IEEE World Forum on Internet of Things, pp. 70–75, 2016.
- [59] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, K. Taylor, “IoT-Lite: A Lightweight Semantic Model for the Internet of Things, in Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress, 2016 Intl IEEE Conferences, pp. 90–97, 2016.
- [60] FOAF ontology, Available at: <http://xmlns.com/foaf/spec/>. Last accessed at 12/10/2017.
- [61] Στάμου, Γ., 2015. “Αναπαράσταση οντολογικής γνώσης και συλλογιστική”. [ηλεκτρ. βιβλ.] Αθήνα:Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Available at: <http://hdl.handle.net/11419/4225>.
- [62] Jena framework. Available at: <https://jena.apache.org/>. Last accessed at 02/09/2017.
- [63] J. Broekstra, A. Kampman, and F. Van Harmelen, “Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema,” in Proceedings of the First International Semantic Web Conference (ISWC 2002) (I. Horrocks and J. Hendler, eds.), pp. 54–68, Springer, 2002.
- [64] B. Motik, “Reasoning in Description Logics Using Resolution and Deductive Databases”. PhD thesis, Karlsruhe Institute of Technology, 2006.
- [65] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A Practical OWL-DL Reasoner,” Web Semantics: Science, Services and Agents on the World Wide Web, vol. 5, pp. 51–53, June 2007.
- [66] D. Tsarkov and I. Horrocks, “FaCT++ Description Logic Reasoner: System Description,” in Automated Reasoning (U. Furbach and N. Shankar, eds.), pp. 292–297, Springer, 2006.
- [67] V. Haarslev, K. Hidde, R. Moller, and M. Wessel, “The RacerPro knowledge representation and reasoning system,” in Semantic Web Journal, vol. 3, no. 3, pp. 267-277, 2012.
- [68] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, “Hermit: An OWL 2 Reasoner,” Journal of Automated Reasoning, vol. 53, pp. 245–269, May 2014.
- [69] C. Maternaghan, K. J. Turner, “Policy conflicts in home automation,” in Computer Networks, pp. 2429-2441, 2013.
- [70] A. F. Layouni, L. Logrippo, K. J. Turner, “Conflict Detection in Call Control Using First-Order Logic Model Checking”, in Feature Interactions in Software and Communications Systems, 2007.
- [71] M. Nakamura, H. Igaki, Y. Yoshimura and K. Ikegami, “Considering Online Feature Interaction Detection and Resolution for Integrated Services in Home

- Network System,” in *Feature Interactions in Software and Communications Systems*, 2009.
- [72] C. Maternaghan, K. J. Turner, “Pervasive computing for home automation and telecare,” in *Pervasive Communications Handbook*, pp. 1-25, 2011.
- [73] R. Camacho, P. Carreira, I. Lynce, S. Resendes, “An ontology-based approach to conflict resolution in Home and Building Automation Systems,” in *Expert Systems with Applications*, vol. 41, pp. 6161-6173, 2014.
- [74] Chieh-Jan Mike, B. F. Karlsson, N. D. Lane, F. Zhao, J. Zhang, Z. Pan, Z. Li, Y. Yu, “SIFT: Building an Internet of Safe Things,” in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, pp. 298-309, 2015.
- [75] N. Tillmann and J. de Halleux, “Pex - White Box Test Generation for .NET,” In *Proceedings of Tests and Proofs (TAP’08)*, Springer Verlag, 2008.
- [76] N. R. Jennings and M. Wooldridge. 1998. Applications of intelligent agents. In *Agent technology*, Nicholas R. Jennings and Michael J. Wooldridge (Eds.). Springer-Verlag New York, Inc., Secaucus, NJ, USA 3-28.
- [77] M. Luck, P. McBurney, C. Preist, “A Manifesto for Agent Technology: Towards Next Generation Computing”, *Autonomous Agents and Multi-Agent Systems*, Kluwer Academic Publishers, Volume 9, Issue 3, pp 203-252, 2004.
- [78] Foundation for Intelligent Physical Agents (FIPA). Available at: <http://www.fipa.org/>. Last accessed at 22/08/2017.
- [79] B. N. Schilit and M. M. Theimer, “Disseminating active map information to mobile hosts,” *IEEE Networks*, vol. 8, no. 5, pp. 22-32, 1994.
- [80] G. D. Abowd and E. D. Mynatt, “Charting past, present, and future research in ubiquitous computing,” *ACM Transactions on Computer-Human Interaction*, vol. 7, pp. 29–58, March 2000.
- [81] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, P. Steggles, “Towards a better understanding of context and contextawareness,” *Handheld and ubiquitous computing*. Springer, pp. 304–307, 1999.
- [82] Ö. Yürür, C. H. Liu, Z. Sheng, Victor C. M. Leung, W. Moreno, K. K. Leung, “Context-Awareness for Mobile Sensing: A Survey and Future Directions”, *IEEE Communications Surveys & Tutorials*, Issue 1, pp. 68-93, 2016.
- [83] A. K. Dey, G. D. Abowd and D. Salber, “A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications,” *Human-Computer Interaction*, Vol. 16, pp. 97-166, 2001.
- [84] A. Bayat, M. Pomplun and D. A. Tran, “A Study on Human Activity Recognition Using Accelerometer Data from Smartphones,” *Procedia Computer Science (Elsevier)*, vol. 34, pp. 450-457, 2014.
- [85] M. Popa, “Hand gesture recognition based on accelerometer sensors,” in *7th International Conference on Networked Computing and Advanced Information Management (NCM)*, 2011.

- [86] E. Thammasat and J. Chaicharn, "A simply fall-detection algorithm using accelerometers on a smartphone," in Biomedical Engineering International Conference (BMEiCON), 2012.
- [87] H. M. Ali and Z. S. Alwan, "Car Accident Detection and Notification System Using Smartphone," International Journal of Computer Science and Mobile Computing (IJCSMC), vol. 4, no. 4, pp. 620-635, 2015.
- [88] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. LeGrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway, P. Klasnja, K. Koscher, J. Landay, J. Lester, D. Wyatt, and D. Haehnel, "The mobile sensing platform: An embedded activity recognition system," IEEE Pervasive Computing, vol. 7, no. 2, pp. 32–41, 2008.
- [89] M. S. Kalas and B. F. Momin, "Stress detection and reduction using EEG signals," in International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016.
- [90] H. W. Guo, Y. S. Huang, J. C. Chien, J. S. Shieh, "Short-term Analysis of Heart Rate Variability for Emotion Recognition via a Wearable ECG Device", International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), pp. 262-265, 2015.
- [91] D. Guan, W. Yuan, S. Lee, and Y.-K. Lee, "Context selection and reasoning in ubiquitous computing," in Intelligent Pervasive Computing, pp. 184-187, 2007.
- [92] L. Sanchez, J. Lanza, R. Olsen, M. Bauer, and M. Girod-Genet, "A generic context management framework for personal networking environments," in 3rd Annual International Conference on Mobile and Ubiquitous Systems - Workshops, pp. 1 –8, July 2006.
- [93] M. Perttunen, J. Rieki, and O. Lassila, "Context representation and reasoning in pervasive computing: a review," in International Journal of Multimedia and Ubiquitous Engineering, vol. 4, no. 4, pp. 1–28, 2009.
- [94] P. Nurmi and P. Floree, "Reasoning in context-aware systems," Position Paper, Department of Computer Science, University of Helsinki, 2004.
- [95] M. Saar-Tsechansky and F. Provost, "Handling Missing Values when Applying Classification Models", Journal of Machine Learning Research, pp. 1625-1657, 2007.
- [96] A. Reiss, D. Stricker, "Creating and benchmarking a new dataset for physical activity monitoring", Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments, Article No. 40, 2012.
- [97] T. Huynh and B. Schiele. "Analyzing features for activity recognition", In sOc-EUSAI '05, pp. 159-163, 2005.

- [98] D. Figo, P. C. Diniz, D. R. Ferreira, J. M. P. Cardoso, "Preprocessing techniques for context recognition from accelerometer data", *Personal and Ubiquitous Computing*, pp. 645-662, 2010.
- [99] I. Goodfellow, Y. Bengio, A. Courville, "Deep Learning (Adaptive Computation and Machine Learning series)," MIT Press, ch. 1, pp. 1-26, 2016.
- [100] R. Raina, A. Battle, H. Lee, B. Packer, A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *proceedings of the 24th International Conference on Machine Learning*, pp. 759-766, 2007.
- [101] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," in *Proceedings of the IEEE*, pp. 2278-2324, November 1998.
- [102] The MNIST database of handwritten digits <http://yann.lecun.com/exdb/mnist/>, last accessed 10/09/2017.
- [103] The ImageNet database. <http://image-net.org/>. Last accessed 10/09/2017.
- [104] I. Goodfellow, Y. Bengio, A. Courville, "Deep Learning (Adaptive Computation and Machine Learning series)," MIT Press, ch. 10, pp. 330-372, 2016.
- [105] deeplearning.ai, "Coursera course on Convolutional Neural Networks". <https://www.coursera.org/learn/convolutional-neural-networks/>. Last accessed at 7/11/2017.
- [106] Y. Zhou and R. Chellappa, "Computation of optical flow using a neural network," In *IEEE International Conference on Neural Networks*, pp. 71-78, 1988.
- [107] M. Lin, Q. Chen, S. Yan, "Network In Network," In *International Conference on Learning Representations*, 2014.
- [108] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," In *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.
- [109] T. Plötz, N. Y. Hammerla, and P. Olivier, "Feature learning for activity recognition in ubiquitous computing," In *Proceedings of the Twenty-Second IJCAI*, vol. 2, pp. 1729-1734, AAAI Press, 2011.
- [110] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," in *Neural computation*, vol. 18, no. 7, pp. 1527-1554, 2006.
- [111] G. E. Hinton, R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks", *Science*, vol. 313, pp. 504-507, 2006.
- [112] C. Vollmer, H.-M. Gross, and J. P. Eggert, "Learning features for activity recognition with shift-invariant sparse coding", In *Artificial Neural Networks and Machine Learning—ICANN*, pp. 367-374, 2013.
- [113] H. Lee, C. Ekanadham, and A.Y. Ng., "Sparse deep belief net model for visual area V2," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 20, 2008.



- [114] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4277-4280, 2012.
- [115] M. Zeng, L. T Nguyen, B. Yu, O. J Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," In *MobiCASE*, pp. 197-205, IEEE, 2014.
- [116] Tutorial on Autoencoders. <http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/>. Last accessed 27/10/2017.
- [117] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks," in *Proceedings of International Conference on Web-Age Information Management*, pp. 298-310, 2014.
- [118] C. A. Ronao, Sung-Bae Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, pp. 235-244, October 2016.
- [119] C. A. Ronao and Sung-Bae Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors," In *Neural Information Processing*, pp. 46-53, Springer, 2015.
- [120] C. A. Ronao and Sung-Bae Cho, "Evaluation of deep convolutional neural network architectures for human activity recognition with smartphone sensors", in *Proceedings of the KIISE Korea Computer Congress*, pp. 858-860, 2015.
- [121] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, S. Krishnaswamy, "Deep Convolutional Neural Networks On Multichannel Time Series For Human Activity Recognition," in *IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 3995-4001, 2015.
- [122] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, Hwee-Pink Tan, "Deep Activity Recognition Models with Triaxial Accelerometers," *The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence*, 2015.
- [123] W. Jiang, Z. Yin, "Human Activity Recognition using Wearable Sensors by Deep Convolutional Neural Networks", in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1307-1310, 2015.
- [124] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning internal representations by error propagation," In *Parallel Distributed Processing*, vol. 1, chapter 8. MIT Press, Cambridge, pp. 318-362, 1986.
- [125] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory," in *Neural Computation*, vol 9, no. 8, pp. 1735-1780, 1997.

- [126] F. J. Ordóñez and D. Roggen, “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition,” *Sensors*, vol. 16, no. 1, pp. 1-25, January 2016.
- [127] Protégé Open-source Ontology Editor. Available at <http://www.epc-rfid.info/>. Last accessed at 27/03/2016.
- [128] E. C. Eze, S. Zhang, E. Liu, “Vehicular ad hoc networks (VANETs): Current state, challenges, potentials and way forward,” 20th International Conference on Automation and Computing, pp. 176-181, 2014.
- [129] Electronic Product Code. Available at <http://www.epc-rfid.info/>. Last accessed at 27/03/2016.
- [130] P. Kasnesis, C. Patrikakis and I. S. Venieris, “Collective domotic intelligence through dynamic injection of semantic rules,” in the Proceedings of the IEEE International Conference on Communications, pp. 592 – 597, 2015.
- [131] JAVA Agent DEvelopment Framework (JADE). Available at: <http://jade.tilab.com/>. Last accessed at 22/03/2017.
- [132] A. Chortaras, G. Stamou, A. Stafylopatis, “Learning Ontology Alignments Using Recursive Neural Networks,” in *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, Springer, vol. 3697, pp. 811-816, 2005.
- [133] A. Poulouvasilis, G. Papamarkos, P. T. Wood, “Event-Condition-Action Rule Languages for the Semantic Web,” *Workshop on Semantic Web and Databases*, pp. 855-864, 2003.
- [134] S. Russell and P. Norvig, “Artificial Intelligence A Modern Approach,” Third Edition, Prentice-Hall press, ch. 9, pp. 322-357, 2010.
- [135] C. Del Vescovo, B. Parsia, U. Sattler and T. Schneider, “The Modular Structure of an Ontology: Atomic Decomposition,” in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 2232-2237, 2011.
- [136] Processing framework. Available at: <https://processing.org/>. Last accessed at 02/09/2017.
- [137] RabbitMQ. Available at: <https://www.rabbitmq.com/>. Last accessed at 02/09/2017.
- [138] D. Anguita, A. Ghio, L. Oneto, X. Parra and J. L. Reyes-Ortiz, “A Public Domain Dataset for Human Activity Recognition Using Smartphones,” in *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013*, April 2013.
- [139] A. Reiss and D. Stricker, “Introducing a New Benchmarked Dataset for Activity Monitoring,” in the *16th IEEE International Symposium on Wearable Computers (ISWC)*, 2012.
- [140] A. Reiss, M. Weber, and D. Stricker, “Exploring and Extending the Boundaries of Physical Activity Recognition,” In *IEEE SMC Workshop on Robust Machine Learning Techniques for Human Activity Recognition*, pp. 46-50, 2011.

- [141] P. Kasnesis, C. Patrikakis and I. S. Venieris, "PerceptionNet: A Deep Convolutional Neural Network for Late Sensor Fusion," Intelligent Systems Conference (IntelliSys), under review.
- [142] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," Technical report, arXiv 1212.5701.
- [143] X. G. Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 249-256, 2010.
- [144] K. He, X. Zhang, S. Ren, J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in IEEE International Conference on Computer Vision (ICCV), 2015.
- [145] L.J.P. van der Maaten, G.E. Hinton, "Visualizing High-Dimensional Data Using t-SNE," in Journal of Machine Learning Research, vol. 9, pp. 2579-2605, 2008.
- [146] M. Ermes, J. Pärkkä, J. Mäntyjärvi, I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," in IEEE Transactions on Information Technology in Biomedicine, vol. 12, no. 1, pp. 20-26, January 2008.
- [147] A. Reiss and D. Stricker, "Creating and Benchmarking a New Dataset for Physical Activity Monitoring," in Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments, Article no. 40, June 2012.
- [148] STORM H2020 EU project, Safeguarding Cultural Heritage through Technical and Organisational Resources Management, <http://www.storm-project.eu/>, accessed October 2016.
- [149] Protecting The Cultural Heritage From Natural Disasters, Available at: [http://www.europarl.europa.eu/thinktank/en/document.html?reference=IP-OL-CULT\\_ET\(2007\)369029](http://www.europarl.europa.eu/thinktank/en/document.html?reference=IP-OL-CULT_ET(2007)369029) , Last accessed at 03/09/2017.
- [150] M. A. Al-garadi, K. D. Varathan, S. D. Ravana, E. Ahmed, V. Chang, "Identifying the influential spreader in multilayer interaction online social networks", Accepted in Journal of Intelligent & Fuzzy Systems, May 2016.
- [151] Fitbit Charge HR, Available at: <https://www.fitbit.com/eu/chargehr/>. Last accessed at 29/06/2017.
- [152] E. Kovatcheva, R. Nikolov, M. Madjarova, A. Chikalanov, "Internet of Things for Wellbeing – Pilot Case of a Smart Health Cardio Belt", Proceeding of XIII Mediterranean Conference on Medical and Biological Engineering and Computing, pp. 1221-1224, 2013.
- [153] S. Hiremath, G. Yang, K. Mankodiya, "Wearable Internet of Things: Concept, architectural components and promises for person-centered healthcare", Proceedings of 4th International Conference on Wireless Mobile Communication and Healthcare (Mobihealth), pp. 304-307, 2014.

- [154] Nike+ Training Club App, Available at: [http://www.nike.com/us/en\\_us/c/nike-plus/training-app/](http://www.nike.com/us/en_us/c/nike-plus/training-app/). Last accessed at 29/06/2017.
- [155] Adidas miCoach App, Available at: <http://www.adidas.com/us/micoach/>. Last accessed at 29/06/2017.
- [156] Withings Smart Body Analyzer, Available at: <http://www.withings.com/eu/en/products/smart-body-analyzer/>. Last accessed at 29/06/2017.
- [157] Siafu Open Source Context Simulator, Available at: <http://siafusimulator.org/>. Last accessed at 29/06/2017.
- [158] J. Pennington, R. Socher, C. D. Manning, “GloVe: Global Vectors for Word Representation,” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.
- [159] D. E. Charilas, A. D. Panagopoulos, A survey on game theory applications in wireless networks, *Computer Networks* 54(18), pp. 3421-3430, 2010.
- [160] J. Mitola, “Cognitive radio architecture evolution,” *Proceedings of the IEEE*, Vol. 97, No. 4, pp. 626–641, April 2009.
- [161] Facebook API. Available at: <https://developers.facebook.com/>. Last accessed at 31/08/2017.
- [162] A. Segall, “Distributed network protocols,” *IEEE Transactions of Information Theory*, vol. IT-29, pp. 23-35, Jan. 1983.
- [163] C. Gkantsidis, M. Mihail and A. Saberi, “Hybrid Search Schemes for Unstructured Peer-to-Peer Networks,” *Proceedings of IEEE Infocom 2005*, Vol. 3, pp. 1526-1537, 2005.
- [164] D. Kogias, K. Oikonomou, and I. Stavrakakis, “Study of randomly replicated random walks for information dissemination over various network topologies”, *Proceedings of Sixth Annual Conference on Wireless on Demand Network Systems and Services (WONS)*, pp. 53-60, February 2-4, 2009.
- [165] Y. Shoham, K. Leyton-Brown, “Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations”, Cambridge University, 2008.
- [166] AlphaGo wins human grandmaster at Go. Available at: <https://www.theguardian.com/technology/2016/mar/15/googles-alphago-seals-4-1-victory-over-grandmaster-lee-sedol>. Last accessed at 30/12/2017.
- [167] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, “Asynchronous Methods for Deep Reinforcement Learning,” in *Proceedings of the 33<sup>rd</sup> International Conference on Machine Learning*, 2016.
- [168] Alexa chat bot. Available at: <https://www.amazon.jobs/en/teams/alexa-machine-learning>. Last accessed at 30/12/2017.
- [169] M. Lewis, D. Yarats, Y. N. Dauphin, D. Parikh, D. Batra, “Deal or No Deal? End-to-End Learning for Negotiation Dialogues,” *arXiv preprint arXiv:1706.05125*, 2017.

- [170] A Das, S Kottur, JMF Moura, S Lee, D Batra, “Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning,” arXiv preprint arXiv:1703.06585, 2017.
- [171] G. Koch, R. Zemel, R. Salakhutdinov, “Siamese Neural Networks for One-shot Image Recognition,” Proceedings of the 32<sup>nd</sup> International Conference on Machine Learning, Lille, France, 2015.
- [172] A. Graves, G. Wayne et al., “Hybrid computing using a neural network with dynamic external memory,” in Nature 538, pp. 471–476, 2016.
- [173] R. Arandjelovic, Andrew Zisserman, “Look, Listen and Learn,” IEEE International Conference on Computer Vision (ICCV), 2017.



## Παράρτημα

### A.1 Παρουσίαση κώδικα για Αναγνώριση Ανθρώπινης Δραστηριότητας (PerceptionNet)

```
In [1]: from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation
from keras.layers import Convolution2D, MaxPooling2D, GlobalAveragePooling2D
from keras.optimizers import Adadelta
from keras.utils import np_utils
from keras.callbacks import Callback, EarlyStopping, ModelCheckpoint
from sklearn.metrics import precision_recall_fscore_support, confusion_matrix

import scipy.io
import numpy as np

Using Theano backend.
Using gpu device 0: GeForce GTX TITAN X (CNMeM is enabled with initial size: 95.0% of memory, cuDNN 5105)
```

```
In [2]: batch_size = 64
nb_classes = 6

# input signal dimensions
img_rows, img_cols, img_channels = 6, 128, 1

data = scipy.io.loadmat('/home/adminis/notebooks/DeepLearningTutorials/code/ucl_all.mat')
#subjects = [302,619,907,1201,1512,1848,2212,2566]
```

```
In [3]: class LossHistory(Callback):
def on_train_begin(self, logs={}):
    self.losses = []
    self.val = []

def on_batch_end(self, batch, logs={}):
    self.losses.append(logs.get('loss'))
    self.val.append(logs.get('val'))
```

```
In [4]: def prepare_data(data):
X_train1 = data['train_X'][:5868, :, :, :].reshape([5868, 1, 6, 128])
X_train2 = data['train_X'][6243:6625, :, :, :].reshape([382, 1, 6, 128])
X_train = np.concatenate((X_train1, X_train2), 0)

y_train1 = np.asarray(data['y_train'], dtype=int)[:5868, :] - 1
y_train2 = np.asarray(data['y_train'], dtype=int)[6243:6625, :] - 1
y_train = np.concatenate((y_train1, y_train2), 0)

X_val1 = data['train_X'][5868:6243, :, :, :].reshape([375, 1, 6, 128])
X_val2 = data['train_X'][6625, :, :, :].reshape([727, 1, 6, 128])
X_val = np.concatenate((X_val1, X_val2), 0)

y_val1 = np.asarray(data['y_train'], dtype=int)[5868:6243, :] - 1
y_val2 = np.asarray(data['y_train'], dtype=int)[6625, :] - 1
y_val = np.concatenate((y_val1, y_val2), 0)

X_test = data['test_X'][:, :, :, :].reshape([data['test_X'].shape[0], 1, 6, 128])
y_test = np.asarray(data['y_test'], dtype=int) - 1

X_means = np.zeros(6)
X_stds = np.zeros(6)

for i in range(6):
    X_means[i] = np.mean(X_train[:, :, :, i])
    X_stds[i] = np.std(X_train[:, :, :, i])
    X_train[:, :, :, i] = (X_train[:, :, :, i] - X_means[i]) / X_stds[i]
    X_val[:, :, :, i] = (X_val[:, :, :, i] - X_means[i]) / X_stds[i]
    X_test[:, :, :, i] = (X_test[:, :, :, i] - X_means[i]) / X_stds[i]

# convert class vectors to binary class matrices
Y_train = np_utils.to_categorical(y_train, nb_classes).astype('float32')
Y_test = np_utils.to_categorical(y_test, nb_classes).astype('float32')
Y_val = np_utils.to_categorical(y_val, nb_classes).astype('float32')

return X_train, Y_train, X_val, Y_val, X_test, Y_test, y_test, y_val
```

```
In [5]: def myModel():
        #create the deep learning model
        model = Sequential()

        model.add(Convolution2D(48, 1, 15, border_mode='same', init='he_uniform', input_shape=(img_channels, img_rows, img_cols)))
        model.add(Activation('relu'))
        model.add(MaxPooling2D(pool_size=(1, 2)))
        model.add(Dropout(0.40))

        model.add(Convolution2D(96, 1, 15, border_mode='same', init='he_uniform'))
        model.add(Activation('relu'))
        model.add(MaxPooling2D(pool_size=(1, 2)))
        model.add(Dropout(0.40))

        model.add(Convolution2D(96, 3, 15, subsample=(3,1), border_mode='same', init='he_uniform'))
        model.add(Activation('relu'))
        model.add(GlobalAveragePooling2D())
        model.add(Dropout(0.40))

        model.add(Dense(nb_classes, init='he_uniform'))
        model.add(Activation('softmax'))

        return model
```

```
In [6]: #train the network
def trainHAR(ensemble, train=False):
    acc = []
    X_train, Y_train, X_val, Y_val, X_test, Y_test, y_test, y_val = prepare_data(data)

    # let's train the model using adadelata.
    adadelata = Adadelata(lr=1, rho=0.95, epsilon=1e-08)

    # compile the model
    model = myModel()
    model.compile(loss='categorical_crossentropy',
                  optimizer=adadelata,
                  metrics=['accuracy'])

    history = LossHistory()
    earlyStopping = EarlyStopping(monitor='loss', patience=100, verbose=0, mode='auto')
    checkpoint = ModelCheckpoint('ucl_'+str(ensemble)+'_best.h5', monitor='val_acc', verbose=1, save_best_only=True, mode='max')
    hist = model.fit(X_train, Y_train,
                    batch_size=batch_size,
                    nb_epoch=300,
                    validation_data=(X_test, Y_test),
                    shuffle=True,
                    callbacks=[earlyStopping, history, checkpoint],
                    verbose=2)

    scipy.io.savemat('ucl_'+str(ensemble)+'_best_losses.mat', {'history': hist.history})

    model.load_weights('ucl_'+str(ensemble)+'_best.h5')

    # get accuracy per subject
    print(np.mean(model.predict_classes(X_test[:302, :, :]) == y_test[:302, :].T))
    print(np.mean(model.predict_classes(X_test[302:619, :, :]) == y_test[302:619, :].T))
    print(np.mean(model.predict_classes(X_test[619:907, :, :]) == y_test[619:907, :].T))
    print(np.mean(model.predict_classes(X_test[907:1201, :, :]) == y_test[907:1201, :].T))
    print(np.mean(model.predict_classes(X_test[1201:1521, :, :]) == y_test[1201:1521, :].T))
    print(np.mean(model.predict_classes(X_test[1521:1848, :, :]) == y_test[1521:1848, :].T))
    print(np.mean(model.predict_classes(X_test[1848:2212, :, :]) == y_test[1848:2212, :].T))
    print(np.mean(model.predict_classes(X_test[2212:2566, :, :]) == y_test[2212:2566, :].T))
    print(np.mean(model.predict_classes(X_test[2566, :, :]) == y_test[2566, :].T))

    #overall test accuracy
    test_acc = np.mean(model.predict_classes(X_test[:, :, :]) == y_test[:, :].T)
    print("Test accuracy: ", np.mean(model.predict_classes(X_test[:, :, :]) == y_test[:, :].T))

    #validation accuracy
    val_acc = np.mean(model.predict_classes(X_val[:, :, :]) == y_val[:, :].T)
    print("Validation accuracy: ", np.mean(model.predict_classes(X_val[:, :, :]) == y_val[:, :].T))

    acc.append(val_acc)
    acc.append(test_acc)

    return acc
```



```
In [7]: def ensembleConvNet():
    acc = np.zeros((10,2))
    X_train, Y_train, X_val, Y_val, X_test, Y_test, y_test, y_val = prepare_data(data)

    model = myModel()

    ensemble_preds = np.zeros((2947,6,10))
    sub2 = X_test[:302, :, :]
    sub4 = X_test[302:619, :, :]
    sub9 = X_test[619:907, :, :]
    sub10 = X_test[907:1201, :, :]
    sub12 = X_test[1201:1521, :, :]
    sub13 = X_test[1521:1848, :, :]
    sub18 = X_test[1848:2212, :, :]
    sub20 = X_test[2212:2566, :, :]
    sub24 = X_test[2566, :, :]

    en_sub2 = np.zeros((sub2.shape[0], 6, 10))
    en_sub4 = np.zeros((sub4.shape[0], 6, 10))
    en_sub9 = np.zeros((sub9.shape[0], 6, 10))
    en_sub10 = np.zeros((sub10.shape[0], 6, 10))
    en_sub12 = np.zeros((sub12.shape[0], 6, 10))
    en_sub13 = np.zeros((sub13.shape[0], 6, 10))
    en_sub18 = np.zeros((sub18.shape[0], 6, 10))
    en_sub20 = np.zeros((sub20.shape[0], 6, 10))
    en_sub24 = np.zeros((sub24.shape[0], 6, 10))

    for i in range(10):
        model.load_weights('ucl_'+str(i)+'_best.h5')

        ensemble_preds[:, :, i] = model.predict(X_test)
        # get accuracy per subject
        en_sub2[:, :, i] = model.predict(sub2)
        en_sub4[:, :, i] = model.predict(sub4)
        en_sub9[:, :, i] = model.predict(sub9)
        en_sub10[:, :, i] = model.predict(sub10)
        en_sub12[:, :, i] = model.predict(sub12)
        en_sub13[:, :, i] = model.predict(sub13)
        en_sub18[:, :, i] = model.predict(sub18)
        en_sub20[:, :, i] = model.predict(sub20)
        en_sub24[:, :, i] = model.predict(sub24)

        #overall test accuracy
        test_acc = np.mean(model.predict_classes(X_test[:, :, i])==y_test[:, :, i].T)
        print("Test accuracy: ", test_acc)

        #validation accuracy
        val_acc = np.mean(model.predict_classes(X_val[:, :, i])==y_val[:, :, i].T)
        print("Validation accuracy: ", val_acc)

        acc[i,0] = val_acc
        acc[i,1] = test_acc
        print("")
    Y_pred = np.argmax(np.mean(ensemble_preds,2),1)
    print("Ensemble", np.mean(np.argmax(np.mean(ensemble_preds,2),1)==y_test[:, :, i].T))
    print("Ensemble per subject:")
    print("Ensemble 2", np.mean(np.argmax(np.mean(en_sub2,2),1)==y_test[:302, :, i].T))
    print("Ensemble 4", np.mean(np.argmax(np.mean(en_sub4,2),1)==y_test[302:619, :, i].T))
    print("Ensemble 9", np.mean(np.argmax(np.mean(en_sub9,2),1)==y_test[619:907, :, i].T))
    print("Ensemble 10", np.mean(np.argmax(np.mean(en_sub10,2),1)==y_test[907:1201, :, i].T))
    print("Ensemble 12", np.mean(np.argmax(np.mean(en_sub12,2),1)==y_test[1201:1521, :, i].T))
    print("Ensemble 13", np.mean(np.argmax(np.mean(en_sub13,2),1)==y_test[1521:1848, :, i].T))
    print("Ensemble 18", np.mean(np.argmax(np.mean(en_sub18,2),1)==y_test[1848:2212, :, i].T))
    print("Ensemble 20", np.mean(np.argmax(np.mean(en_sub20,2),1)==y_test[2212:2566, :, i].T))
    print("Ensemble 24", np.mean(np.argmax(np.mean(en_sub24,2),1)==y_test[2566, :, i].T))

    precision_avg, recall_avg, f_score_avg, avg = precision_recall_fscore_support(y_test, Y_pred, average='weighted')
    conf_mat = confusion_matrix(y_test, Y_pred)

    return acc, precision_avg, recall_avg, f_score_avg, avg, conf_mat
```

```
In [8]: accuracy ,precision_avg, recall_avg, f_score_avg, avg, conf_mat = ensembleConvNet()

2912/2947 [=====>] - ETA: 0s('Test accuracy: ', 0.97522904648795383)
1088/1102 [=====>] - ETA: 0s('Validation accuracy: ', 0.99092558983666057)

2912/2947 [=====>] - ETA: 0s('Test accuracy: ', 0.97522904648795383)
1088/1102 [=====>] - ETA: 0s('Validation accuracy: ', 0.99001814882032668)

2912/2947 [=====>] - ETA: 0s('Test accuracy: ', 0.97081778079402781)
1102/1102 [=====>] - 0s
('Validation accuracy: ', 0.99183303085299457)

2912/2947 [=====>] - ETA: 0s('Test accuracy: ', 0.96199524940617576)
1088/1102 [=====>] - ETA: 0s('Validation accuracy: ', 0.99183303085299457)
```

```

2947/2947 [=====] - 0s
('Test accuracy: ', 0.9667458432304038)
1102/1102 [=====] - 0s
('Validation accuracy: ', 0.99274047186932846)

2947/2947 [=====] - 0s
('Test accuracy: ', 0.96640651510010178)
1088/1102 [=====.] - ETA: 0s('Validation accuracy: ', 0.99092558983666057)

2944/2947 [=====.] - ETA: 0s('Test accuracy: ', 0.96742449949100784)
1056/1102 [=====.] - ETA: 0s('Validation accuracy: ', 0.99092558983666057)

2912/2947 [=====.] - ETA: 0s('Test accuracy: ', 0.96301323379708181)
1056/1102 [=====.] - ETA: 0s('Validation accuracy: ', 0.99183303085299457)

2947/2947 [=====] - 0s
('Test accuracy: ', 0.96606718696979976)
1056/1102 [=====.] - ETA: 0s('Validation accuracy: ', 0.99092558983666057)

2912/2947 [=====.] - ETA: 0s('Test accuracy: ', 0.96844248388191378)
1088/1102 [=====.] - ETA: 0s('Validation accuracy: ', 0.99092558983666057)

('Ensemble', 0.97251442144553779)
Ensemble per subject:
('Ensemble 2', 0.98013245033112584)
('Ensemble 4', 0.98107255520504733)
('Ensemble 9', 0.9097222222222221)
('Ensemble 10', 0.92517006802721091)
('Ensemble 12', 0.96250000000000002)
('Ensemble 13', 1.0)
('Ensemble 18', 0.99450549450549453)
('Ensemble 20', 0.98022598870056499)
('Ensemble 24', 1.0)
    
```

In [9]: `print(precision_avg, recall_avg, f_score_avg)`

(0.97306192606000574, 0.97251442144553779, 0.97243874010127329)

```

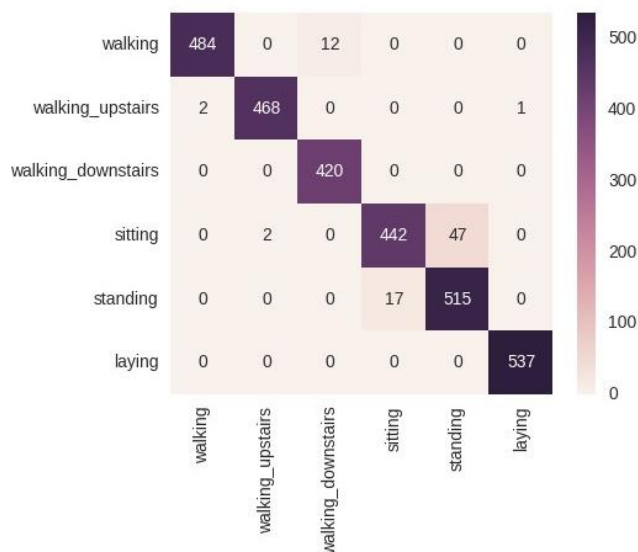
In [10]: import pandas as pd
import seaborn as sn
import matplotlib.pyplot as plt
%matplotlib inline

classes = ['walking', 'walking_upstairs', 'walking_downstairs', 'sitting', 'standing', 'laying']

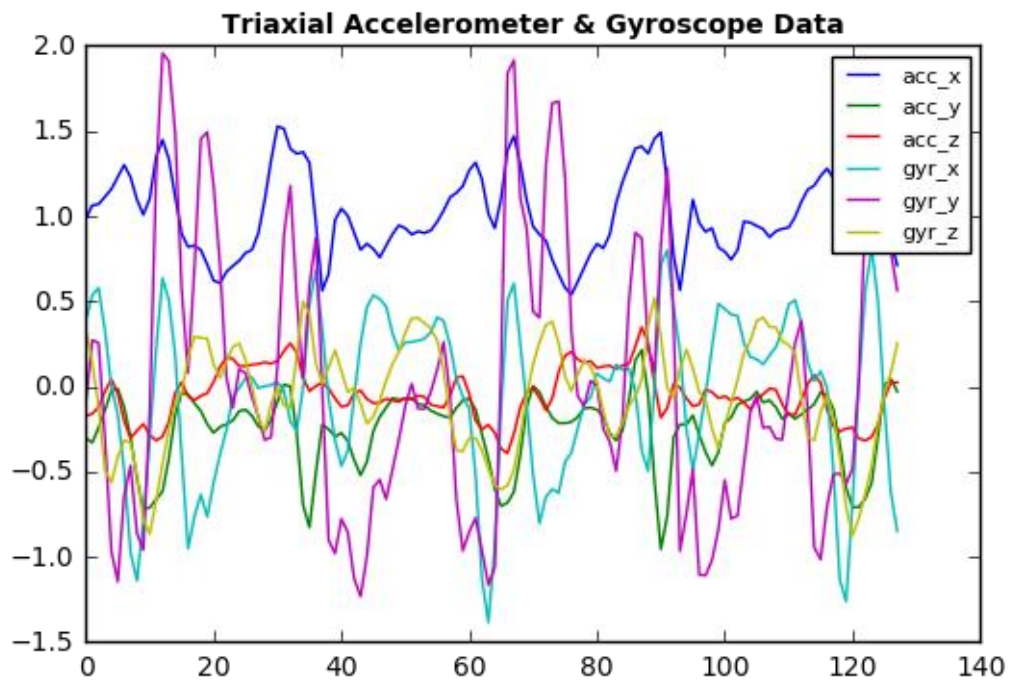
df_cm = pd.DataFrame(conf_mat.astype(int), index = [i for i in classes],
                    columns = [i for i in classes])
plt.figure(figsize = (5,4))

sn.heatmap(df_cm, annot=True,fmt='g')
    
```

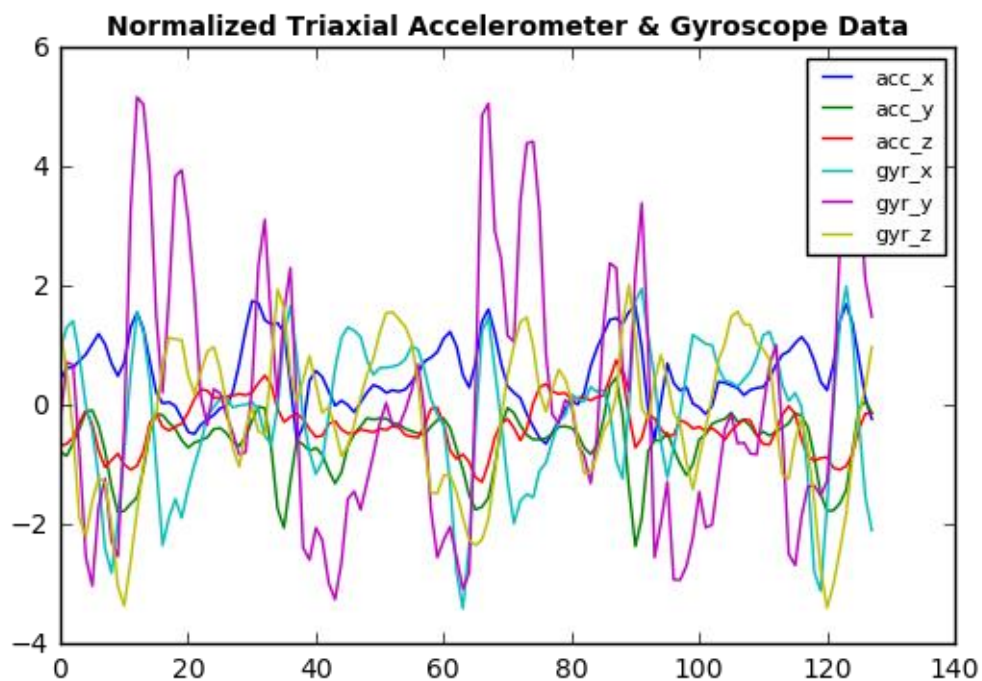
Out[10]: <matplotlib.axes.\_subplots.AxesSubplot at 0x7fee394d9c50>



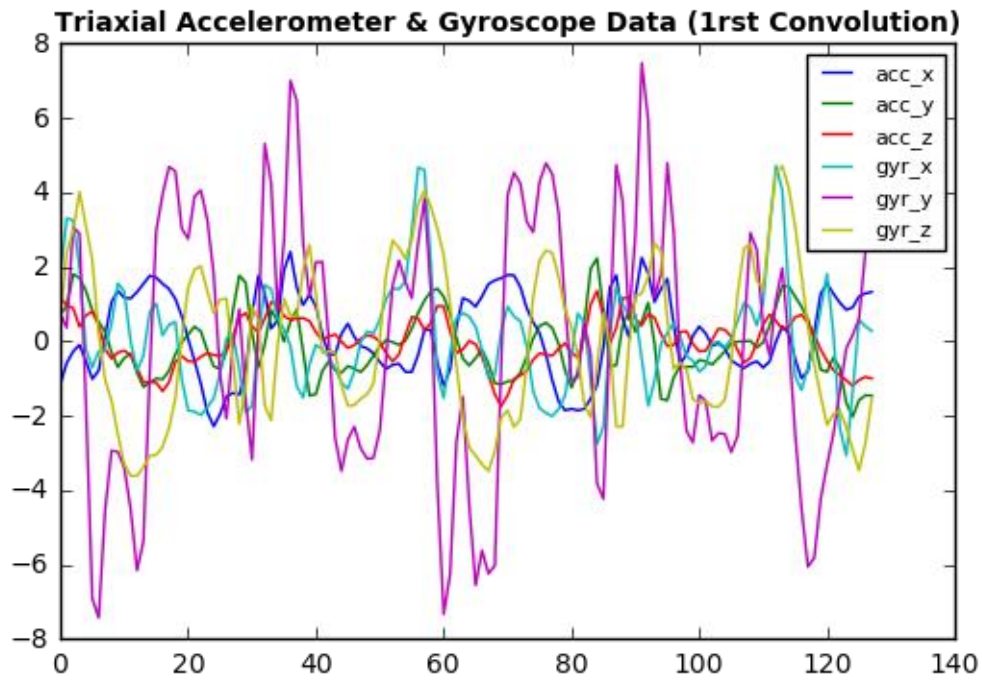
## A.2 Γραφικές παραστάσεις των σημάτων κίνησης κατά τη διάρκεια επεξεργασίας τους



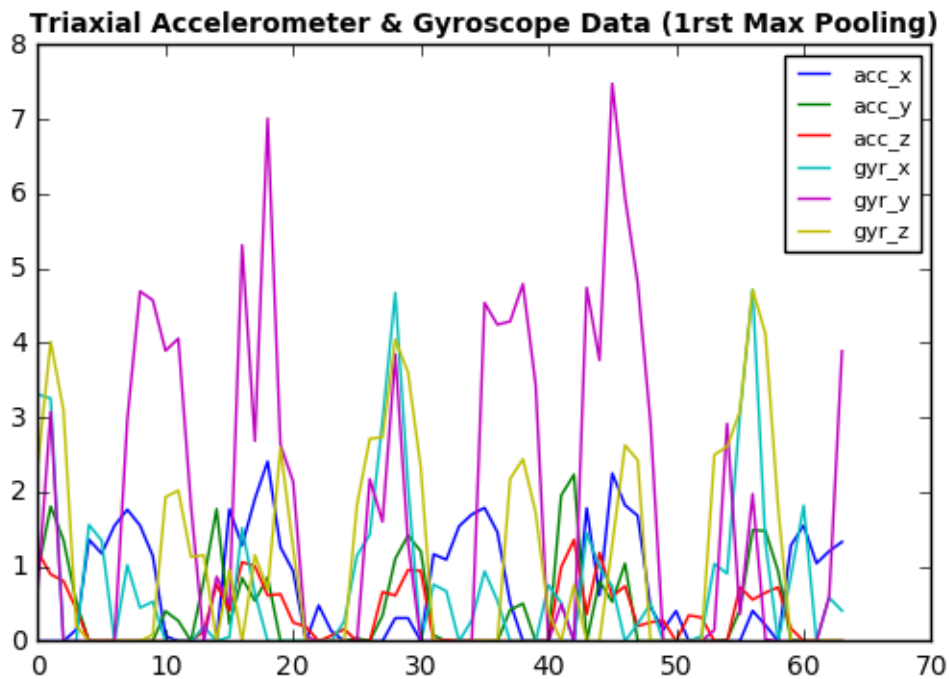
Εικόνα 1: Ακατέργαστα σήματα τριαξονικού επιταχυνσιόμετρου και γυροσκοπίου



Εικόνα 2: Κανονικοποιημένα σήματα τριαξονικού επιταχυνσιόμετρου και γυροσκοπίου

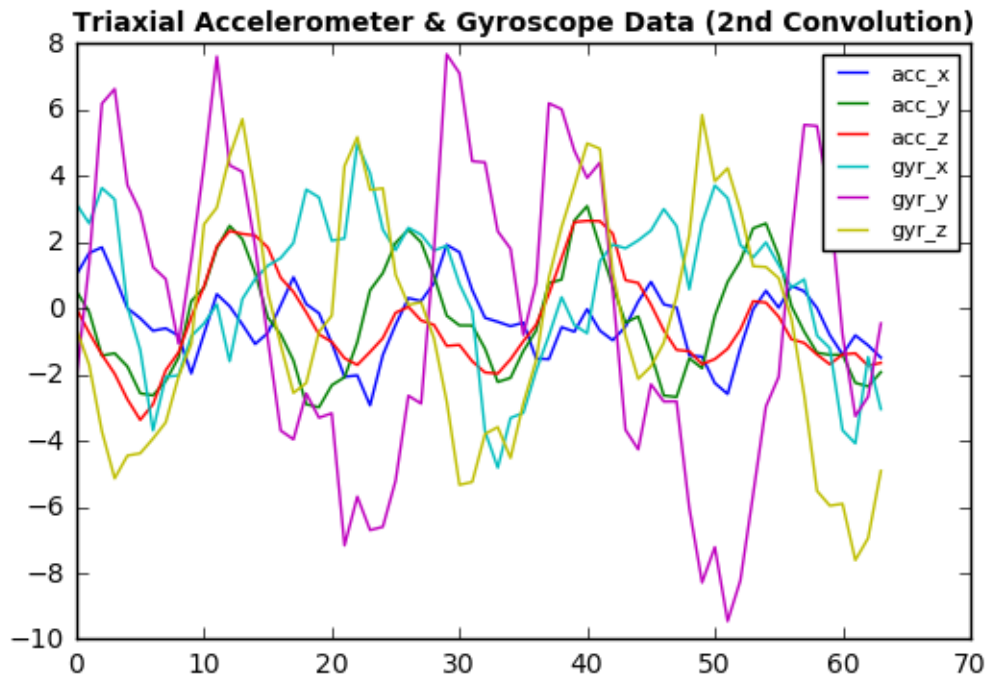


Εικόνα 3: Σήματα τριαξονικού επιταχυνσιόμετρου και γυροσκοπίου μετά την πρώτη μονοδιάστατη συνέλιξη

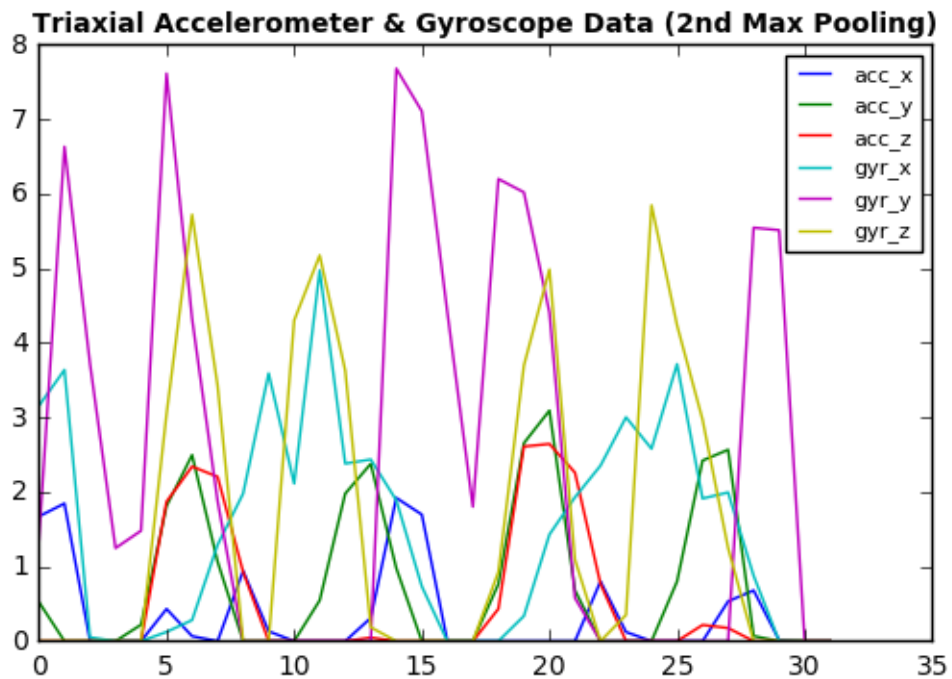


Εικόνα 4: Σήματα τριαξονικού επιταχυνσιόμετρου και γυροσκοπίου μετά την πρώτη υποδειγματοληψία μέγιστης τιμής

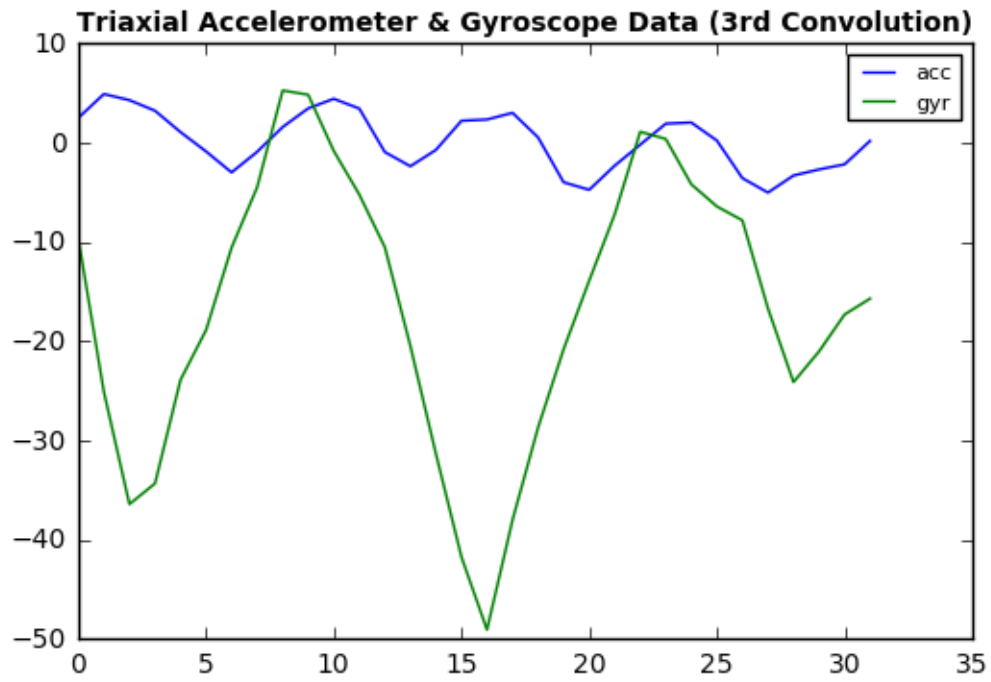




Εικόνα 5: Σήματα τριαξονικού επιταχυνσιόμετρου και γυροσκοπίου μετά τη δεύτερη μονοδιάστατη συνέλιξη



Εικόνα 6: Σήματα τριαξονικού επιταχυνσιόμετρου και γυροσκοπίου μετά τη δεύτερη υποδειγματοληψία μέγιστης τιμής



Εικόνα 7: Σήματα τριαξονικού επιταχυνσιόμετρου και γυροσκοπίου μετά τη δισδιάστατη συνέλιξη

## Δημοσιεύσεις

### Διεθνή περιοδικά

- P. Kasnesis, C. Patrikakis and I. S. Venieris, "*Changing Mobile Data Analysis through Deep Learning*", in the IEEE IT Professional Mobile Data Analytics, Vol. 19, pp. 17–23, 2017.
- P. Kasnesis, C. Patrikakis, D. Kogias, L. Toumanidis and I. S. Venieris, "*Cognitive Friendship and Goal Management for the Social IoT*", in the Computers & Electrical Engineering, Elsevier, Vol. 58, pp. 412–428, 2017.
- A. Kapsalis, P. Kasnesis, I. S. Venieris, D. I. Kaklamani and C. Patrikakis, "*A Cooperative Fog Approach for Effective Workload Balancing*", IEEE Cloud Computing, Vol. 4, pp. 36-45, 2017.
- D. Meridou, M.-E. Papadopoulou, P. Kasnesis, C. Patrikakis, G. Lamprinakos, A. Kapsalis, I. S. Venieris and D. I. Kaklamani, "*The Health Avatar: Privacy-aware monitoring and management of health and lifelog data*", IEEE IT Professional Wearable Computing, pp. 20-27, 2015.

### Κεφάλαια σε βιβλία

- D. T. Meridou, M.-E. Ch. Papadopoulou, A. P. Kapsalis, P. Kasnesis, C. Z. Patrikakis, I. S. Venieris, D.-T. I. Kaklamani, "*Improving Quality of Life with the Internet of Everything*", In Beyond the Internet of Things: Everything Interconnected. J. M. Batalia, G. Mastorakis, C. X. Mavromoustakis and E. Pallis eds. Springer, pp. 377-408, 2017.

### Πρακτικά συνεδρίων

- P. Kasnesis, C. Patrikakis and I. S. Venieris, "*PerceptionNet: A Deep Convolutional Neural Network for Late Sensor Fusion*," Intelligent Systems Conference (IntelliSys), accepted.
- P. Kasnesis, L. Toumanidis, D. Kogias, C. Patrikakis and I. S. Venieris, "*ASSIST: An Agent-Based SIoT Simulator*", in the Proceedings of the 2016 IEEE World Forum on Internet of Things, pp. 353-358, 2016.
- A. Kapsalis, P. Kasnesis, P. C. Theofanopoulos, P. Gkonis, C. Lavranos, D. I. Kaklamani, I. S. Venieris and G. Kyriacou, "*A Cloud Platform for Classification and Resource Management of Complex Electromagnetic Problems*", In Proceedings of the 7th IC3K 2015, pp. 388 – 393, 2015.
- D. Meridou, C. Patrikakis, M.-E. Papadopoulou, P. Kasnesis and I. S. Venieris, "*Serving the Needs of Goal Oriented Scenarios through the Deployment of an*

*Intelligent Enterprise Service Bus: A Welfare Use Case", Smart Factory and Industry 4.0, 2015.*

- D. Meridou, A. Kapsalis, P. Kasnesis, C. Patrikakis, I. S. Venieris and D. I. Kaklamani, *"An Event-driven Health Service Bus", in proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare - "Transforming healthcare through innovations in mobile and wireless technologies", (MOBIHEALTH 2015), 2015.*
- P. Kasnesis, C. Patrikakis and I. S. Venieris, *"Collective domotic intelligence through dynamic injection of semantic rules", in the Proceedings of the IEEE International Conference on Communications, pp. 592 – 597, 2015.*