

NATIONAL TECHNICAL UNIVERSITY OF ATHENS

 $School \ of \ Rural \ and \ Surveying \ and \ Geoinformatics$

Engineering

PHOTOGRAMMETRY LABORATORY

Activity Recognition from Visual Cues in and beyond the Visual Spectrum

Doctorate Thesis

Bakalos, G., Nikolaos

Athens, September 2022



NATIONAL TECHNICAL UNIVERSITY OF ATHENS School of Rural, Surveying and Geoinformatics Engineering Photogrammetry Laboratory

Αναγνώριση δραστηριοτήτων από οπτικές πηγές εντός και πέραν του ορατού φάσματος με χρήση βαθιά μηχανικής μάθησης.

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Νικόλαος, Γ., Μπάκαλος

Συμβουλευτική Επιτροπή : Αναστάσιος, Δ., Δουλάμης Θεοδώρα, Βαρβαρίγου Ανδρέας Γεωργόπουλος,

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 20η Σεπτεμβρίου 2022.

Αναστάσιος Δουλάμης Αναπληρωτής Καθηγητής Ε.Μ.Π.

.....

Ανδρέας Γεωργόπουλος Καθηγητής Ε.Μ.Π.

.....

Θεοδώρα Βαρβαρίγου Καθηγήτρια Ε.Μ.Π.

.....

Χαράλαμπος Ιωαννίδης Καθηγητής Ε.Μ.Π.

.....

..... Βασίλειος Βεσκούκης Αναπ. Καθηγητής Ε.Μ.Π.

Κωνσταντίνος Καράντζαλος Αναπ. Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2022

.....

Αθανάσιος Βουλόδημος Επικ.Καθηγητής Ε.Μ.Π.

.....

3

.....

Nikolaos Georgiou Bakalos

Διδάκτωρ Αγρονόμος Τοπογράφος Μηχανικός και Μηχανικός Γεωπληροφορικής Copyright © Nikolaos Georgiou Bakalos, 2022.

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 τον συγγραφέα.

Abstract

Activity recognition from optical cues is an arduous task that has recently received a lot of attention in the research community, due to the performance of deep learning architectures in the analysis of such kinds of data. However, these analyses have to take into account both the types of input data as well as statistical particularities and a priori knowledge over the types of activities captured. This dissertation focuses on the development of deep machine learning methods to classify actions recorded in datasets consisting of capturings inside and outside the visible spectrum. Two main application scenarios are studied.

The first application scenario includes recordings of traditional dance choreographies, where the dataset consists of a predefined set of actions (motion primitives), i.e. the steps that compose the specific dance choreography. The problem then takes the form a mutli-class classification task. Two deep learning classifiers are presented. For the data outside the visual spectrum, in this case recordings of infrared depth sensors, an optimised Long Short Term Memory (LSTM) neural network is presented. This classifier manages to capture both short-term dependencies, by using a short memory window before its input layer, as well as take into account non-causality during classification, by using the bidirectional variant of LSTM networks. For the data inside the visible spectrum, a hybrid architecture is presented. This architecture puts into use the feature extraction capabilities of Convolutional Neural Networks (CNN), as well as the ability of LSTM networks to map temporal correlations. Autoregressive and Moving Average capabilities are added to the architecture, while an adaptive weight control scheme is also employed. Finally, for the first application scenario, a tensor based classifier is presented that manages to classify choreographic motion primitives with similar performance, while requiring significantly less trainable parameters, allowing for increased performance even when a small set of training data is available.

The second application scenario focuses on datasets where there is no a priori knowledge of the actions captured. Instead we study ways to "map" the normal state, and employ techniques for binary classification of the normal and the abnormal state. Initially, a supervised approach is presented, employing an adaptive NARMA filter, based on a CNN architecture. Data fusion from other sensors is also used to inform the classification step and increase performance. Additionally, unsupervised techniques, based on convolutional autoencoders are employed. Finally, a stack autoencoder method is presented where the feature extraction of convolutional spatiotemporal autoencoders is used in combination with a tensor-based autoencoder to model the normal state in datasets with large numbers of actions, and then perform outlier detection.

Keywords

Deep learning, computer vision, analysis of visual cues, supervised and unsupervised learning, adaptive neural network architectures, data fusion, tensor-based learning

Περίληψη

Η αναγνώριση δραστηριότητας από οπτικές ενδείξεις είναι μια επίπονη εργασία που πρόσφατα έχει λάβει μεγάλη προσοχή στην ερευνητική κοινότητα, λόγω της απόδοσης αρχιτεκτονικών βαθιάς μηχανικής μάθησης στην ανάλυση τέτοιων ειδών δεδομένων. Ωστόσο, αυτές οι αναλύσεις πρέπει να λαμβάνουν υπόψη τόσο τους τύπους των δεδομένων εισόδου όσο και τις στατιστικές ιδιαιτερότητες και την εκ των προτέρων γνώση σχετικά με τους τύπους των δραστηριοτήτων που καταγράφονται. Αυτή η διατριβή εστιάζει στην ανάπτυξη μεθόδων βαθιάς μηχανικής μάθησης για την ταξινόμηση ενεργειών που καταγράφονται σε σύνολα δεδομένων που αποτελούνται από καταγραφές εντός και εκτός του ορατού φάσματος. Μελετώνται δύο βασικά σενάρια εφαρμογής.

Το πρώτο σενάριο εφαρμογής περιλαμβάνει καταγραφές παραδοσιακών γορογραφιών, όπου το σύνολο δεδομένων αποτελείται από ένα προκαθορισμένο σύνολο ενεργειών (motion primitives), δηλαδή τα βήματα που συνθέτουν τη συγκεκριμένη χορογραφία. Το πρόβλημα στη συνέχεια παίρνει τη μορφή μιας εργασίας ταξινόμησης πολλαπλών κλάσεων. Παρουσιάζονται δύο ταξινομητές βαθιάς μάθησης. Για τα δεδομένα εκτός οπτικού φάσματος, σε αυτή την περίπτωση εγγραφές αισθητήρων βάθους υπερύθρων, παρουσιάζεται ένα βελτιστοποιημένο νευρωνικό δίκτυο Μακροπρόθεσμης Μνήμης (LSTM). Αυτός ο ταξινομητής καταφέρνει να συλλάβει και τις βραχυπρόθεσμες εξαρτήσεις, γρησιμοποιώντας ένα παράθυρο σύντομης μνήμης πριν από το επίπεδο εισόδου του, καθώς και να λάβει υπόψη τη μη αιτιότητα κατά την ταξινόμηση, χρησιμοποιώντας την αμφίδρομη παραλλαγή των δικτύων LSTM. Για τα δεδομένα εντός του ορατού φάσματος, παρουσιάζεται μια υβριδική αρχιτεκτονική. Αυτή η αρχιτεκτονική χρησιμοποιεί τις δυνατότητες εξαγωγής χαρακτηριστικών των Συνελικτικών Νευρωνικών Δικτύων (CNN), καθώς και την ικανότητα των δικτύων LSTM να χαρτογραφούν χρονικές συσχετίσεις. Δυνατότητες Autoregressive και Moving Average προστίθενται στην αρχιτεκτονική, ενώ χρησιμοποιείται επίσης ένα προσαρμοστικό σύστημα ελέγχου των βάρων του νευρωνικού. Τέλος, για το πρώτο σενάριο εφαρμογής, παρουσιάζεται ένας ταξινομητής βασισμένος σε τανυστή που καταφέρνει να ταξινομήσει χορογραφικές κινήσεις με παρόμοια απόδοση, ενώ

απαιτεί σημαντικά λιγότερες εκπαιδεύσιμες παραμέτρους, επιτρέποντας αυξημένη απόδοση ακόμη και όταν είναι διαθέσιμο ένα μικρό σύνολο δεδομένων εκπαίδευσης.

Το δεύτερο σενάριο εφαρμογής εστιάζει σε σύνολα δεδομένων όπου δεν υπάρχει εκ των προτέρων γνώση των ενεργειών που καταγράφονται. Αντίθετα, μελετήσαμε τρόπους για να «χαρτογραφήσουμε» την κανονική κατάσταση και χρησιμοποιήσαμε τεχνικές για δυαδική ταξινόμηση της κανονικής και της ανώμαλης κατάστασης. Αρχικά, παρουσιάζεται μια εποπτευόμενη προσέγγιση, που χρησιμοποιεί ένα προσαρμοστικό φίλτρο NARMA, βασισμένο σε αρχιτεκτονική CNN. Η σύντηξη δεδομένων από άλλους αισθητήρες χρησιμοποιείται επίσης για να ενημερώσει το βήμα ταξινόμησης και να αυξήσει την απόδοση. Επιπλέον, χρησιμοποιούνται τεχνικές χωρίς επίβλεψη, που βασίζονται σε συνελικτικούς αυτοκωδικοποιητές. Τέλος, παρουσιάζεται μια μέθοδος αυτόματου κωδικοποιητή στοίβας όπου η εξαγωγή χαρακτηριστικών των συνελικτικών χωροχρονικών αυτοκωδικοποιητών χρησιμοποιείται σε συνδυασμό με έναν αυτόματο κωδικοποιητή που βασίζεται σε τανυστική μάθηση για να μοντελοποιήσει την κανονική κατάσταση σε σύνολα δεδομένων με μεγάλο αριθμό ενεργειών και στη συνέχεια να εκτελέσει ανίχνευση ακραίων τιμών.

Λέξεις Κλειδιά:

Βαθιά μηχανική μάθηση, όραση υπολογιστών, ανάλυση οπτικών δεδομένων, επιβλεπόμενη και μη επιβλεπόμενη μάθηση, προσαρμοστικές δομές νευρωνικών δικτύων, σύμμειξη δεδομένων, τανυστική μάθηση

Table of Contents

Abstract
Keywords
Περίληψη7
Λέξεις Κλειδιά:
Table of Figures
Table of Tables
1. Introduction
2. Previous Works
2.1. Supervised learning in motion primitive recognition
2.2. Supervised learning in the identification of outlier actions
2.2.1. Localization using channel state information from WiFi
2.2.2. Cyber security of sensors, PLC and SCADA
2.2.3. Fusion across multiple data modalities
2.3. Unsupervised learning in the identification of outlier actions
2.4. Contribution
3. Identification of motion primitives: data outside the visible spectrum
3.1. Problem Formulation and Notation
3.2. Bi-directional Long-Range Dependence
3.3. Bayesian Optimization
3.4. Experimental Evaluation
3.5. Conclusions
4. Identification of motion primitives: data inside the visible spectrum
4.1. Mathematical Formulation

	4.1.	1.	Network Weight Adaptation	42
4	.2.	Perf	ormance Evaluation	43
	4.2.	1.	Dataset Description	43
4	.3.	Con	clusions	45
5.	Iden	tifica	tion of motion primitives: Space-Time Tensor Based Neural Networks	for
trai	ning u	ınder	Small Sample Settings	46
5	5.1.	Prob	olem Formulation	46
5	5.2.	Data	a Processing	49
5	5.3.	Spac	ce-Time Domain Tensor Based Neural Network	49
	5.3.	1.	CSP Neural Network Layer	50
	5.3.2.		Tensor Fusion Operation	51
	5.3.	3.	Tensor Based Neural Network	52
5	5.4.	Perf	ormance Evaluation	53
	5.4.	1.	Performance Evaluation Against State of the Art Methods	59
5	5.5.	Con	clusions	61
6.	Sup	ervise	ed Approach: Simulating scenarios in multimodal datasets	63
6	5.1.	Mod	lelling input data modalities	65
	6.1.	1.	Visual modality: RGB & thermal camera streams for vision-based detect 65	ion
	6.1.2	2.	WiFi signal reflection modality for human intrusion detection	66
	6.1.3. analysis		ICS sensing modality: PLC and SCADA data for cyber-physical atta 68	ıck
	6.1.4 char	4. nnels	Multimodal data fusion from visual, WiFi reflection and ICS sensing in 68	put
6	5.2.	The	proposed adaptive deep learning model for cyber-physical event detection	. 69

	6.2.1.	Tapped Delay Line Convolutional Neural Network (TDL-CNN)	
	6.2.2.	Adaptive TDL-CNN	
6.	.3. Exp	erimental evaluation	74
	6.3.1.	Experiment setup	74
	6.3.2.	Results	
6.	4. Con	clusions	80
7.	Unsuper	vised Approach: Fall detection in optical and thermal datasets	
7.	1. Stac	ked Convolutional Autoencoders for Feature Extraction	
7.	2. Eva	luation	
	7.2.1.	Dataset Description	
	7.2.2.	Model Training	
7.	.3. Con	clusions	
8.	Unsuper	vised Approach: Outlier detection in datasets including	numerous
simu	ultaneous	actionss	
8.	1. Intra	a-Inter property encoding	
	8.1.1.	Property Representations	
	8.1.2.	Intra Property Encoding using spatiotemporal autoencoders	100
	8.1.3.	Inter Property Encoding using tensor-based unsupervised learning	; 100
	8.1.4.	Unsupervised Tensor _based Learning	
		Chsupervised Tensor –based Learning	
8.	2. The	Rank-1 Canonical Decomposition of Network Parameters	
8.	2. The 8.2.1.	Rank-1 Canonical Decomposition of Network Parameters The Learning Algorithm	104 105
8. 8.	 The 8.2.1. Exp 	Rank-1 Canonical Decomposition of Network Parameters The Learning Algorithm	104 105 105
8. 8. 8.	 The 8.2.1. Exp Con 	Rank-1 Canonical Decomposition of Network Parameters The Learning Algorithm erimental evaluation	104 105 105 109
8. 8. 8. 9.	 The 8.2.1. Exp Conclusi 	Rank-1 Canonical Decomposition of Network Parameters The Learning Algorithm erimental evaluation clusions	104 105 105 109 110

Table of Figures

Figure 1. Summary of implemented methods presented in this dissertation
Figure 2. The architecture of a bi-directional feedforward neural network with time delay
line filter able to model non-causal relationships among the choreographic pose primitives
Figure 3. (a) The architecture of the memory cell for the LongShort Term Memory (LSTM)
network, (b) Bidirectional LSTM unfolded in time
Figure 4. The choreographic primitives for Sirtos (3-Beat)
Figure 5. The effect of the memory length (e.g., the time delay line filter) on the accuracy
criterion for different classifiers
Figure 6. The performance of the proposed BOBi-LSTM network for pose identification at
different frame indices. Ground truth data are also depicted
Figure 7. Performance of BOBi-LSTM model on pose identification versus conventional
LSTM structures
Figure 8. The proposed adaptive convolutionally enriched LSTM network with
AutoRegressive and Moving Average capabilities
Figure 9. The main choreographic primitives of the Syrtos (3-beat) dance sequence 43
Figure 10. The effect of background subtraction on the choreographic modelling
performance
Figure 11. The effect of the memory window in the classification performance
Figure 12. Kinect II skeletal capturing system (vvvv.org/documentation/kinect)
Figure 13. The proposed CSP layer and the tensor fusion operation. Parameter N stands for
the number of skeleton joints

Figure 14. Propagation of information through the layers of the tensor-based neural network Figure 15. Average classification accuracy and F1 score of a tensor-based neural network Figure 16. Average classification accuracy and F1 score of a tensor-based neural network with one tensor contraction layer, for M = 24, and for different values of parameter T 55 Figure 17. Average classification accuracy and F1 score of a tensor-based neural network Figure 18. Average classification accuracy and F1 score of a tensor-based neural network with different number of tensor contraction layers (parameter K) for M = 24 and T = 11. 58 Figure 20. A high-level overview of a framework to tackle diverse types of cyber and Figure 21. Schematic overview of human presence detection mechanism from WiFi Figure 23. The effect of autoregressive – moving average behavior on the classification performance (F1-score) in the case of multimodal data fusion using (a) shallow learning classifiers and (b) deep learning ones. Short memory corresponds to considering 30 previous Figure 24. The effect of autoregressive – moving average behavior on the classification performance (F1-score) in case that (a) data from only the visual modality are used, and (b) data from only WiFi signal reflection or ICS sensing compared to the respective behavior on Figure 25. Performance metrics for CNN and the proposed TDL-CNN, as well as their adaptive versions. Applying the adaptive scheme improves the classification performance in

Figure 31. Structure of the Convolutional Kernel Operator
Figure 34. The human-sized dummy that was used during the test throws
Figure 35. Test throws during the data collection experiments. The free fall (a)-(d) of the
human dummy from different shooting angles (positive event), and various other objects
such as (e) plastic bags and (f) bottles (negative event)
Figure 36. The RGB optical sensor, which was used during the data acquisition experiments
to monitor the test throws of the human dummy, mounted on the building
Figure 37. The four locations of the building where the optical sensor was placed, during
the data acquisition experiments91
Figure 38. Autoencoder ROC Curve Sideways Camera
Figure 39. Autoencoder ROC Curve Top Camera
Figure 40. Comparative analysis unsupervised vs supervised approach for both capturing
angles
Figure 42. Performance of multi-autoencoder approach96
Figure 43. Proposed twofold architectures for abnormal event detection
Figure 44. The tensor based learning algorithm adopted in the unsupervised tensor based
network
Figure 45. Our approach for abnormal event detection as outliers of normal space
partitioning by the unsupervised tensor learning105
Figure 47. Performance difference between different number of k in the Shangai and Avenue
Dataset
Figure 48. Performance difference between different levels of noise in the video stream,
Avenue Dataset
Figure 49. Captured abnormalities and system response (Avenue Dataset). Axis x presents
the frame batch while axis y represents the average reconstruction error. Above the detected
abnormalities the annotated ground-truth data is presented109

Table of Tables

Table 1. A brief description of the dances recorded from Kinect-II
Table 2. Performance evaluation and comparisons
Table 3. Comparative Performance Evaluation of the proposed method with other classifiers
Table 4. Distribution of annotated samples between classes for each dance (performer)54
Table 5. Projections of tensor objects when they propagated through tensor contraction
layers (TCL) and the ranks of the tensor regression layer (TRL)
Table 6. Performance comparison in terms of average classification accuracy and F1 score
against LSTM and BOBi LSTM models60
Table 7. Classification performance metrics for experiments using a single data modality
(visual, WiFi signal reflection, ICS sensing). Four different classification methods have been
examined
Table 8. Classification performance metrics and execution times (per 100 frames) for
experiments using fusion of all modalities (visual, WiFi signal reflection, and ICS sensing).
Table 9. Performance of the unsupervised autoencoder approach for the different positions
of the camera
Table 10. Performance of a supervised classifier for the different positions of the camera94
Table 11. Abnormal Behavior detection based on frame level AUC on the Avenue and
Shanghai tech datasets106
Table 12. Research difference summary of our work with [98]. 107

1. Introduction

Action recognition on optical data is considered an arduous task that requires the analysis of high dimensional input signals both in the temporal and spatial field. It includes the analysis of movements identified over sets of sensors and it typically relies on numerous methods spanning from digital signal processing to the extraction of salient characteristics from the raw data in order to feed a machine learning model. Nowadays, deep learning methods have become the gold standard in such analyses, since architectures such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have established state of the art performances in the extraction of characteristics from raw data that drive classification or regression tasks.

The rapidly increasing computational power of the past decades has allowed the development of complex models that are able to explain complicated physical mechanisms. Moreover, innovations in data capturing, storage and retrieval technologies, e.g. novel sensor networks, big data database architectures, has created a plethora of data sources that can be used for the training of deep learning models, and transform monitoring and control tasks over large and complicated infrastructures. Consequently, modern activity recognition problems are characterized by complexity. Also, since real-world systems often evolve under transient conditions, the signals received tend to exhibit various forms of non-stationarity. As for mathematical models, they can be categorized in many different ways. They can be linear or non-linear, static or dynamic, continuous or discrete over time, deterministic or contemplative. The model chosen for the description of a system depends on the system under study, on whether the operation of the system is known or not, as well as on the purpose of implementing the model. This dissertation proposes techniques for identifying activities based on deep machine learning, comparing them under specific application scenarios. More specifically, algorithms were developed for two main sub-cases of the activity recognition problem.

The first case concerns the study of visual data sets, in which there is prior knowledge of the actions that were recorded. In this case, an application scenario was chosen which concerns the analysis of recordings of Greek traditional dances with the aim of creating models for the automatic annotation of the primitive steps contained in the performance. The problem in this case takes the form of a multidisciplinary classification problem, for the solution of which supervised machine learning techniques were used.

The second case concerns the study of data sets, for which there is no prior knowledge about the actions they contain. In this case the problem turns into a problem of outlier detection. The use of supervised machine learning using test datasets and the attempt to generalize these models to real-world conditions were studied. Furthermore, the use of non-supervised machine learning models for the analysis of benchmarking datasets was studied.

In both sub-cases, sets of visual data both inside and outside the visible spectrum were analyzed, as well as tensor-based learning techniques for limiting training parameters in order to maximize the performance of the models under development. The rest of this dissertation is structured in the following way:

Chapter 2 provides the theoretical background for the development of models of deep machine learning. First, the relevant literature is presented. Specifically, after extensive research, the most important literature references related to the methods for modeling both the first and the second subproblem are described.

Chapter 3 presents the proposed supervised machine learning method for the annotation of motion primitives in choreography analysis, in data sets outside the visible spectrum. Specifically, the problem analyzes recordings similar to the previous chapter, but using recording sensors in the infrared spectrum. The use of these sensors enables the rapid extraction of the skeletal structure of the person being recorded, and the transformation of the problem into a multidimensional time series analysis problem. Use of Long Short-Term Memory Networks (LSTM) networks is recommended. LSTM networks are considered suitable for time series data modeling as they are "memory" networks and past inputs affect future forecasts. The proposed model is based on bidirectional LSTM networks, introducing the property of non-causality and thus achieving greater accuracy in the model.

Chapter 4 presents the proposed supervised machine learning method for the annotation of motion primitives in choreography analysis, in data sets within the visible spectrum. The method is based on the development of a deep learning machine model that extends the well-known Convolutional Neural Networks (CNN) to simulate the behavior of a NARMA (Non-Linear Autoregressive Moving Average) model.

In Chapter 5 the analysis focused on the use of tensor-based learning to limit training parameters. Specifically, a new deep neural network is introduced based on a tensor network model capable of automatically processing and correlating spatio-temporal information from different sources and discovering appropriate patterns for assigning inputs to the desired outputs. This is a general space-time learning machine, which can be useful for a variety of time series analysis applications, such as human behavior recognition, moving object analysis, radar signals, audio processing, etc. Here the research focused on the classification of human posture using three-dimensional skeletal information in a manner similar to that of chapter 3.

Chapter 6 presents the use of supervised machine learning techniques in data sets in which there is no prior information about the activities recorded. Experimental models with representative activities were created and an algorithm based on convolutional deep neural networks that can analyze inputs both inside and outside the visible spectrum was developed.

Chapter 7 presents an unsupervised learning method for analyzing sets inside and outside the optical spectrum. Specifically, a convolutional spatiotemporal autoencoder was developed which has the ability to model activities that describe the "normal" state. The recognition of statistical endpoints is achieved by analyzing the decoder reconstruction error.

Chapter 8 extends the use of autoencoders to the analysis of data sets containing a large number of concurrent activities. The use of tensor-based learning and in particular an automatic tensor autoencoder is done in combination with convolutional autoencoders with the aim of developing a model that can represent the normal state even when it contains complex movements and a large number of people.

Finally, Chapter 9 presents the summary and contribution of the doctoral dissertation as well as the conclusions that emerged, framed by ideas for future research in the field.

2. Previous Works

2.1. Supervised learning in motion primitive recognition

The approaches regarding the decoding of the human kinesiology are distinguished into: i) supervised and ii) unsupervised categories and mainly take as input RGB images and depth/skeletal data. The decoding and explanation of the human activity by observing only individual representative postures and their temporal variations in a sequence of video frames has been a challenge in the field of Computer Vision and ICH. In the literature, many applications are proposed regarding the human activity indexing [1], pose identification [2], action prediction [3], emotion recognition [4] and background subtraction [5].

In [6], an unsupervised approach for understanding activity by means of its most grained temporal constituents is proposed. In [7], a spatio-temporal decomposition of kinesiology sequences based on a hierarchically modification of the SMRS algorithm is introduced. In [8], an approach to model videos using dense sampling with feature tracking is introduced. Moreover, descriptors combine motion information and trajectory shape for action localization and video retrieval purposes. In [9], features from shapes and optical flow are combined for classification purposes. Hidden Markov Model (HMM) is adopted using multi-frame averaging method for background extraction.

Deep learning methods have been shown to outperform previous state-of-the-art machine learning techniques in several fields, with computer vision being one of the most prominent cases [10]. In [11], a CNN-based feature extraction approach that extracts the local dependency and scale invariant characteristics is proposed. In [12], the problem of human activity recognition by combining multiple vision cues of RGB-D sensor is proposed. In [13], a deep video classification model with competitive performance is introduced. Specifically, this model embeds separate spatial and temporal recognition streams based on ConvNets. In [14], a novel three-stream CNN embedding deep learnt single frame, optical flow and maximizing significant difference and independence (MSDI) features is introduced. The architecture is implemented in the spatial and temporal domain. In [15], a

method for human action recognition from depth data and skeletal data using deep CNNs is proposed. This architecture used two action representations and three CNNs channels in order to maximize the feature extraction procedure. In [16], a fully automated behaviour understanding through visual cues in industrial environments is proposed constructing features from spatial and temporal dimensions. In [17], the authors propose a flexible Deep CNN framework, a Deep Event Network (DevNet), that detects high-levels events and localizes spatial-temporal evidences. This framework takes into consideration keyframes of videos as input data detecting the event of interest by aggregating the CNN features.

Background subtraction (BS) is a challenging task in Computer Vision field especially in real-time application scenarios. BS methods are distinguished into the following categories: i) Foreground detection (FS) and ii) Background detection (BS). During the first category, a comparison process between the current frame and the background model is carried out. In the second category, the obtained images are analysed, updating the background model learned at the initialization step. In general, the BS field comprises the basic, statistical, fuzzy and neural techniques. The BS algorithms are used to detecting moving objects in video sequences from the difference between the current frame and a reference frame. In [18], the authors present a real-time maritime surveillance system based in VAM, background subtraction and an adaptive NN tracker. In [19], a novel background subtraction from video sequences algorithm using deep Convolutional Neural Network is introduced. The proposed approach consists of three processing steps, background model generation, CNN for feature extraction and post-processing. In [20], the authors introduce a region-based Mixture of Gaussians (MoG) for background subtraction in order to handle the sensitivity to dynamic background. In [21], the authors introduce a novel deep background subtraction method by proposing a guided learning methodology that learns a predefined CNN model for each video without pre-labelling process.

2.2. Supervised learning in the identification of outlier actions

Detection of physical intrusion and direct attacks on main infrastructure calls for automatic supervision and immediate identification of suspicious behavior, which can be effected by computer vision tools. These tools aim at exploiting smart video surveillance to detect humans, operating in limited visual conditions, performing just-in-time computation to suggest preventive actions. Computer vision tools that operate outside of the visible spectrum (i.e. thermal sensors) are also gaining traction in this context, because they are not significantly affected by illumination changes [26]. However, such approaches do not capture texture or color information. Vision techniques focus on background and target modelling [27], object tracking [28], target detection [24], activity recognition [29], crowd dynamics and identification of unusual and suspicious behavior [30]. These approaches aim at detecting abnormalities in crowded environments by analyzing actions both on the spatial and temporal scales. Detailed surveys about video-based abnormal activity recognition have been published [31], [32].

2.2.1. Localization using channel state information from WiFi

Several studies have been carried out which leverage the properties of radiofrequency devices to detect a person. Focusing on the device-free solutions, there are techniques based on SDR (Software Defined Radio) devices and custom antenna-arrays, like RF-Capture. One approach is to analyze the Received Signal Strength (RSS) of a wireless signal since the latter undergoes measurable distortions upon the presence of humans or due to human movement [33]. However, RSS is not sufficiently accurate and consistent due to the high variability of these signals [34]. In 2011, a tool based on a COTS WiFi network card has been released [35] which uses an Intel FW modification that allows the upper layers of the protocol to acquire this CSI information used in WiFi devices. Recent studies [25] have shown that analysing the correlation changes over different subcarriers provides a robust and accurate method for detecting human presence with a 99% success rate, even in the case of TTWD (Through the Wall Detection).

2.2.2. Cyber security of sensors, PLC and SCADA

The first works to address the problem of cyber-security in complex infrastructures [36] involved a bank of delay-differential observer systems based on an analytically approximate model of canal hydrodynamics. The method was tested on a class of adversarial scenarios of a generalized fault/attack model. In [22] a modelling framework was developed to characterize the effect of cyber-physical attacks on the hydraulic behavior of water distribution systems. The model identifies the components of the cyber infrastructure (e.g., sensors or PLCs) potentially vulnerable to attacks, determining the exact specifications of an attack (e.g., timing, duration) and simulating it with water system simulation model EPANET. The conclusions is that the same hydraulic response can be obtained through different attack scenarios.

Further, the "BATtle of the Attack Detection Algorithms (BATADAL)" conducted at the EWRI-ASCE conference (https://batadal.net/), extended this work through an algorithm competition for detecting cyber-attacks on a test case study about a water distribution system operated through PLCs and a SCADA. Related results are summarized in [37], while the most successful method of the competition was based on understanding of the physical behavior of the water distribution system operation, combined with an anomaly detection technique [23]. Finally, in [38] an augmented graph assembly is employed between sensors to actuators, which is then tested to sustain malicious attacks to water distribution systems prior to failure.

2.2.3. Fusion across multiple data modalities

As for works involving fusion of multiple data modalities, the majority of existing methods pertains to combined use of RGB and thermal (or hyperspectral) sensors for computer vision. Fusion of thermal and RGB sensors, e.g., has been used to create semantically enriched visual information structures [39]. This is also the case in [40] where discrete wavelet transform (DWT) is used, combined with a SVM for feature classification. In the field of person detection using WiFi reflection, fusion with visual and other (e.g. inertia) sensors has been used to a limited extent, to increase the accuracy of location estimation and eliminate problems arising from signal oscillation and other interfering issues [41]. An example of visual and sensor data fusion is [42], where RGB data are fused with

laser sensor and GPS data. However, to our knowledge, there are no previous works in the literature considering the fusion of visual and WiFi reflection data, with ICS sensing data in the context of water infrastructure monitoring.

2.3. Unsupervised learning in the identification of outlier actions

Abnormal event detection in video surveillance, a process to detect specific frames containing an anomaly, has been drawn a great attention in image processing research mainly due to its advantages in many applications [43] - [46]. Examples include surveillance in industrial environments [44] or critical infrastructures [45] for safety/security and quality assurance, traffic flow management [46] and intelligent monitoring of public places [47].

Some works address abnormal event detection as a multi-class classification problem under a supervised paradigm ([44],[45]). The main, however, limitation of such approaches is that abnormal events sporadically occur in real-world videos. Additionally, what is an abnormal event is vague and tough to model. This means that the distribution of normal versus abnormal events is *severely imbalanced* which result in *low classification performance*. One solution to address this issue is to use semi-supervised learning [48],[49]. However, again the problem of *data imbalance among normal and abnormal cases cannot be handled*. For this reason, the abnormal event detection problem is modeled as *outlier detector*. In particular, the model learns the normality from data samples and then it identifies the abnormal events as the ones which deviate from the normal learnt cases [50] - [52].

In this context, *unsupervised learning* has been applied to handle abnormal event detection [53]-[55]. The methods partition the normal space into coherent clusters in contrast to the outlier-detector models that they use a common global model for the whole normal space. Then, the abnormality is detected as those events which cannot be represented by the normal space. Usually, k-means clustering algorithm is utilized (as in [53])combined with SVM learning. We concentrate on works handling abnormal event detection either as an outlier detection or using deep/ unsupervised learning schemes. Regarding outlier detection, the works of [50], [52], [56]learn dictionary of sub-events, through a training process, and then those events that do not lie in the partitioned sub-space are marked as abnormal ones.

Regarding deep learning, the work of [55] employs convolutional auto-encoders (ConvAE) to learn temporal regularity in videos, while auto-encoders are exploited in [57] to learn feature and reconstruct the input images. Then, one-class Support Vector Machines (SVMs) are used for detecting the abnormal events. The work of [58] introduces a hybrid scheme which aggregates ConvAE with Long Short-Term Memory (LSTM) encoder-decoder. Recently, deep generative models have been applied [59]-[62], modelling, first, the normal space and then, the abnormal is given the difference from the normal one.

Unsupervised learning models are utilized for abnormal event detection. In [63], the anomalies in videos are scored independently of temporal ordering and without any training by simply discriminating between abnormal frames and the normal ones. Other approaches employ tracking algorithms to extract salient motion information which is then classified either as normal or abnormal [64], [65]. However, tracking fails in complex visual scenes of multiple humans' presence.

2.4. Contribution

This dissertation presents techniques developed for the extraction of semantic information in from data capturings inside and outside the visible spectrum. The techniques presented can be broken down in two different categories. A summary of the techniques presented based on the application scenario can be viewed in the figure below.



Application Scenario 1: Frame level classification of motion primitives (choreographic steps) in tradiitional greek dances Application Scenario 2: Outlier detection in multimodal datasets without prior knowledge of the actions recorded

Figure 1. Summary of implemented methods presented in this dissertation

The first category, where there is previous knowledge about the actions captured, includes the development of a Bayesian optimized bi-directional LSTM network [66], of a unimodal hybrid LSTM-CNN architecture with autoregressive and moving average behavior [67] and of a tensor based neural network [68] that can perform similar classification tasks with [66] but with the use of fewer trainable parameters. The second category, where there is no information about the actions captured, presents initially a multimodal CNN architecture extended to showcase autoregressive and moving average behavior, which also includes a novel weight adaptation mechanism to incorporate user feedback [69]. Finally, two unsupervised techniques are presented, based on implementations of deep neural autoencoders. Initially, a set of convolutional autoencoders trained on multiple image properties is used in fall detection scenarios [70]. This implementation is extended to include a tensor-based autoencoder, following a stack autoencoder architecture.

Part 1:

Action recognition in datasets composed from finite predefined motion primitives

3. Identification of motion primitives: data outside the visible spectrum

A special field of computer vision is that of digitization of Cultural Heritage Assets, especially in the field of Intangible Cultural Heritage, such as traditional dance choreographies or other performing arts. This creates a ripe application scenario for developing modelling algorithms based on visual cues that are able to identify the motion primitives that comprise a dance choreography. The algorithms developed should be based on capturings of the performance of a choreography, and also take into account previous folklore studies, that identify specific steps that differentiate this dance from others. This provides us with a priori knowledge on the analysis of the performance, as we know the chain of motion primitives, as well as the ratio of each primitive in the entire performance. This knowledge can be used in the development of a deep learning architecture that allows the automatic classification of dance steps in a capturing of a specific traditional dance.

In this section the infrared mode of a Kinect-II sensor is used as a capturing interface, which translates depth data to *M* 3D skeleton joints. Vector $\vec{J}_k^{\vec{G}} = (x_k^G, y_k^G, z_k^G)$ is the *xyz* coordinates of the *k*-th joint. Superscript *G* indicates the origin of a global coordination system (coincides with the Kinect location). The main limitation of directly processing joints $\vec{J}_k^{\vec{G}}$ is that the choreographic attributes of a dancer are lost, since $\vec{J}_k^{\vec{G}}$ also includes global motion trajectory attributes. For this reason, we transform $\vec{J}_k^{\vec{G}}$ into a local coordination system, the center of which coincides with the center of mass of the dancer. $\vec{J}_k^{\vec{L}} = \vec{J}_k^{\vec{G}} - \vec{c_{cm}}$ Variable c_{cm} is the center of mass of the dancer.

The kinematics of the dancer is modelled using principles of rigid body dynamics [71]. In particular, for every dancer's joint \vec{J}_k^L , the velocity and the acceleration vector are estimated as the first and the second derivative of the *k*th joint position, that is $\vec{u}_k(t) = \frac{d\vec{J}_k^L}{dt}$ and $\vec{\gamma}_k(t) = \frac{d\vec{u}_k}{dt}$.

In this way, the *xyz* coordinates of the velocity and the acceleration are derived as $\overrightarrow{u_k}(t) = (u_k^x, u_k^y, u_k^z)$, and $\overrightarrow{\gamma_k}(t) = (\gamma_k^x, \gamma_k^y, \gamma_k^z)$. It is clear that acceleration actually models the force $F_k(t)$ acting on the *k*-th joint, assuming that mass equals one (m = 1). Therefore, a state vector is derived including all the kinematics properties of a dancer's joint.

$$S_k(t) = \begin{pmatrix} \overrightarrow{j_k^L} \\ \overrightarrow{u_k(t)} \\ \overrightarrow{\gamma_k(t)} \end{pmatrix} = \begin{pmatrix} x_k^L & y_k^L & z_k^L \\ u_k^x & u_k^y & u_k^z \\ \gamma_k^x & \gamma_k^y & \gamma_k^z \end{pmatrix}$$
(1)

In order to include the contribution of all *M* joints, a $3 \cdot M \times 3$ state matrix is constructed.

$$S(t) = (S_1(t), S_2(t), \dots, S_3(t))^T$$
 (2)

3.1. Problem Formulation and Notation

The purpose of the pose choreographic identification is to categorize a dance frame t into a set of L available choreographic primitives. Let us denote as $p_i(t)$ the probability that t frame is assigned to the *i*-th choreographic class. Then, frame t is categorized to the $c^{(t)}$ class

$$\hat{c}(t) = \arg\max_{i \in 1, \dots, L} p_i(t) \quad (3)$$

Let $p(t) = [p_1(t), p_2(t), ..., p_L(t)]$ a vector including all probabilities $p_i(t)$. Usually, $p_i(t)$ is a non-linear relationship of the 3D kimenatics features. Pose identification depends not only on the current dancer's movement, but also on previous and future choreographic primitives. For example, for a particular choreography, a left cross leg is a result of several previous dancer's movements and also implies that, in the future, other pre-determined steps will be followed.

$$p(t) = g(\vec{x}(t), \vec{x}(t-1), \dots, \vec{x}(t-p), \vec{x}(t+1), \dots, \vec{x}(t+p)$$
(4)

In Eq. (4) function $g(\cdot)$ refers to a non-linear vectored value function. Eq. (4) implies that $2 \cdot p + 1$ image frames affect the pose identification at frame *t*.

The main difficulty in implementing Eq. (4) is that function $g(\cdot)$ is actually unknown. However, it has been proven that a feedforward neural network with a Tapped Delay Line (TDL) filter can approximate the $g(\cdot)$ [72] with any degree of accuracy. In this way, the probabilities $p_i(t)$ is a relationship of *L* latent (hidden) state units u_i .

$$\vec{p}(t) = \vec{u}^{T}(t) \cdot \vec{u}$$
$$\vec{u}^{T}(t) = \begin{bmatrix} u_{1}(t) \\ \vdots \\ u_{L}(t) \end{bmatrix} = \begin{bmatrix} \tanh(\overrightarrow{w_{1}}^{T} \cdot \vec{x}(t)) \\ \vdots \\ \tanh(\overrightarrow{w_{L}}^{T} \cdot \vec{x}(t) \end{bmatrix}$$
(5)

In Eq. (5), vector $\vec{x}(t)$ refers to the input data, generated after a vectorization of the matrices S(t + k), with $k = -p, \dots, p$. Moreover, function $tanh(\cdot)$ refers to the hyperbolic tangent function, which is used as an activation function of each hidden neuron unit. Vectors \vec{u} and $\vec{w_i}$ are appropriately estimated by a learning algorithm, usually based on a steepest descent. To better model the non-causal relationships of a choreography, we allow the hidden states units u_i to be related with its previous and future state values.

$$\overrightarrow{u_i}(t) = \tanh(\overrightarrow{w_i}^T \cdot \overrightarrow{x}(t) + \overrightarrow{r_{i,b}}^T \overrightarrow{u_i}(t-1) + \overrightarrow{r_{i,b}}^T \overrightarrow{u_i}(t+1)$$
(6)

Eq. (6) implies a recurrent mechanism within the network states, resulting in a so-called bi-directional recurrent neural network architecture [73]. Figure 2 depicts this architecture used for choreographic pose identification.

3.2. Bi-directional Long-Range Dependence

The main limitation of the aforementioned model is that it is not able to approximate longrange dependencies. However, a dance choreography is composed of repeated patterns, spanned over long time periods. Thus, a bi-directional Long



Figure 2. The architecture of a bi-directional feedforward neural network with time delay line filter able to model non-causal relationships among the choreographic pose primitives



Figure 3. (a) The architecture of the memory cell for the LongShort Term Memory (LSTM) network, (b) Bidirectional LSTM unfolded in time

Short Term Memory (LSTM) network is adopted. LSTMs are of similar structure to the bidirectional recurrent regression models but each node in the hidden layer is replaced by a memory cell, instead of a single neuron [74].

The basic unit of an LSTM is the memory cell. It consists of four components as shown in **Figure 3**. The i) *the forget node*, ii) *the input gate*, iii) *the internal state*, and vi) *the output gate*. Each component non-linearly relates the inner product of the input vectors with appropriate weights, estimated via a training phase. The non-linear activation function adopted for the components is i) the sigmoid denoted as σ and the *tanh*. The *forget gate* throws out (forgets) information from the memory cell to model long-range dependence. *The input node* is the same as a hidden neuron, measuring the contribution of a hidden state to the final classification outcome. *The internal gate* decides if the respective hidden gate is "significant enough" for dance pose identification. Finally, *the output gate* regulates whether the response of the current memory cell is significant enough to contribute to the next cell.

3.3. Bayesian Optimization

A Bayesian strategy is applied for optimally tuning the parameters of the LSTM, in particular the model structure (i.e. number of hidden layers, number of neurons per layer and learning rate). We hereby present the operation of Bayesian Optimization. Let us assume that we have a set of Q different configurations, $D_{1:Q} = \{\vartheta_1 \cdots \vartheta_q\}$. Then, an error is estimated for a given configuration ϑ and an input vector $\vec{x}(t)$, $E(\vec{x}(t),\vartheta)$. Let us now assume that a minimum error E_{min} has been reached over all Q different configurations of the set $D_{1:Q}$. Then, an improvement function is given by

$$I(\vec{x}(t),\vartheta) = max\{0, E_{min} - E(\vec{x}(t),\vartheta)\}$$
(7)

Assuming a probabilistic framework, we take the expectations of the above equation. The target is to estimate a new configuration parameter vector, ϑ^* that further decrease the $I(\cdot)$. Since we do not known the function $I(\cdot)$, one easy way to estimate its respective distribution, using the Bayesian rule.

$$P(E|D_{1:Q}) \propto P(D_{1:Q}|E) \cdot P(E)$$
(8)

Usually P(E) follows a Gaussian distribution and it is proven that $P(D_{1:Q}|E)$ is a Gaussian process of mean value of $\mu(\vartheta)$ and a standard deviation Σ at configuration point ϑ [23].

$$\Sigma = \begin{bmatrix} k(\theta_1, \theta_1) & \dots & k(\theta_1, \theta_Q) \\ \vdots & \dots & \vdots \\ k(\theta_Q, \theta_1) & \dots & k(\theta_Q, \theta_Q) \end{bmatrix}$$
(9)

In case of a new configuration point ϑ^* , the $P(D_{1:Q+1}|E)$ is again a Gaussian process of standard deviation

$$\Sigma = \begin{bmatrix} \Sigma & b \\ b^T & k(\theta_{Q+1}, \theta_{Q+1}) \end{bmatrix} (10)$$

where $b = [k(\vartheta_{Q+1}, \vartheta_1) \cdots k(\vartheta_{Q+1}, \vartheta_Q)]$. Therefore, the new optimal configuration point ϑ^* is given as the integral of the expectation of Eq.(7) and $P(D_{1:Q+1}|E)$ that follows a Gaussian process with known mean and standard deviation.

3.4. Experimental Evaluation

Type of Dance	Description	Main Choreographic Steps
Sirtos (3-Beat)	A Greek folklore dance in a slow three-beat rhythm performed by both women and men.	 Initial Posture (IP); 2) Cross Leg (CL); 3) Initial Posture (IP); 4) Left Leg Up (LLU); 5) Initial Posture (IP); 6) Right Leg Up (RLU);
Sirtos (5-Beat)	A Greek folkloric circular dance performed by both women and men, with a 7/8 musical beat.	1) Initial Posture (IP); 2) Left Leg Back (LLB); 3) Cross Legs (CL); 4) Cross Legs (CL); 5) Cross Legs (CL); 6) Initial Posture (IP); 7) Right Leg Back (RLB);
Kalamatianos	A very popular Greek folk- dance through Peloponnese and the Greek Islands. The tempo is at 7/8 beat.	 Initial Posture (IP); 2) Cross Legs (CL); Cross Legs (CL); 4) Cross Legs (CL); 5) Cross Legs (CL); 6) Initial Posture (IP); 7) Cross Legs Backwards (CLB);
Trehatos	A circle dance, performed by both women and men.	 Initial Posture (IP); 2) Cross Legs (CL); Cross Legs (CL); 4) Cross Legs (CL); 5) Initial Posture (IP); 6) Left Leg Up (LLU); Right Leg Up (RLU); 8) Left Leg Up (LLU); 9) Cross Legs Backwards (CLB);
Enteka	A folkloric dance performed by women and men by at a line	1)Initial Posture (IP); 2) Right Leg Up (RLU); 3) Dancer's Right Turn (DRT); 4) Initial Posture (IP) 5) Dancer's Left Turn (DLT);

Table 1. A brief description of the dances recorded from Kinect-II.

Data Set Description: In our approach, the motion capturing process are funded by the EU project TERPSICHORE [75].



Figure 4. The choreographic primitives for Sirtos (3-Beat).

The data set consists of six Greek folklore dances (Sirtos at 3 beat, Sirtos at 5 beat, Kalamatianos, Trehatos, Enteka). Each dance is performed by three professionals. Several instances (realizations) are considered. **Table 1** presents a brief description of the dances along with the main choreographic primitives, used as categories for pose identification. For example, in Sirtos at 3-beat, we have six main choreographic postures, repeated over time (see **Table 1** and **Figure 4**).

Classification Method	Memory Window	Accuracy	Precision	Recall	F1 Score
	No Memory	45,360%	35,389%	43,293%	38,944%
SVM	5 Frames	55,074%	45,300%	55,988%	50,080%
	10 Frames	70,041%	61,375%	68,982%	64,956%
	No Memory	23,548%	15,037%	19,341%	16,920%
kNN	5 Frames	35,069%	28,007%	39,042%	32,616%
	10 Frames	37,937%	30,004%	40,659%	34,528%
	No Memory	43,047%	32,294%	37,844%	34,850%
FNN 1	5 Frames	61,075%	51,217%	69,281%	58,895%
	10 Frames	65,486%	55,094%	77,066%	64,254%
	No Memory	43,987%	32,753%	37,186%	34,829%
FNN 2	5 Frames	64,835%	54,654%	74,192%	62,941%
	10 Frames	68,426%	58,242%	76,168%	66,009%
	No Memory	61,003%	51,386%	57,725%	54,371%
LSTM	5 Frames	65,558%	55,818%	69,222%	61,802%
	10 Frames	84,189%	75,736%	89,341%	81,978%
	No Memory	62,521%	53,310%	55,449%	54,359%
BOBi-LSTM	5 Frames	71,101%	62,234%	71,737%	66,648%
	10 Frames	85,418%	86,249%	75,868%	80,726%

Table 2. Performance evaluation and comparisons

Table 2 provides a comparison with state of the art methods for choreography modelling. We indicate the performance using methods such as Support Vector Machines (SVMs) and k-Nearest neighbors (kNN) as well as well two different configurations of feedforward neural networks, one with one hidden layer and 10 neurons and one of two hidden layers and 10 neurons per layer. The proposed scheme (BOBi-LSTM) outperforms the compared ones. In the Table, we depict how the classification performance depends on window size (number of frames) fed as input to the classifiers (the effect of the time delay line filter). We observe that as the memory increases the performance improves, but with a decaying improvement ratio. Memory actually acts as a smoothing operation, introducing, however, delay lags in pose identification process. **Figure 5** depicts the accuracy performance versus memory length for different classifiers. The effect of the Bayesian optimization method is depicted in **Figure 6**. As is observed, Bayesian optimization increases the performance of the bi-directional LSTM over all objective criteria. Finally, **Figure 7** depicts indicative outcomes

of the proposed BOBi-LSTM with 10 frame memory in comparison with the ground truth data, over one cycle of Syrtos in 3-beats.



Figure 5. The effect of the memory length (e.g., the time delay line filter) on the accuracy criterion for different classifiers



Figure 6. The performance of the proposed BOBi-LSTM network for pose identification at different frame indices. Ground truth data are also depicted.


Figure 7. Performance of BOBi-LSTM model on pose identification versus conventional LSTM structures.

3.5. Conclusions

In this chapter, we proposed a Bayesian optimized, bi-directional LSTM network for pose characterization of a choreography. Comparisons with other shallow learning classifiers indicates that the proposed scheme is very effective for kinesiology modelling. This is due to the fact that a dance sequence presents i) non-causalities (future steps affect the current performance) and ii) long-range dependencies (several forward or backward steps affect the current dancer's movement).

4. Identification of motion primitives: data inside the visible spectrum

Similarly to the previous chapter, the data used for the development of this deep learning model, are performance captures of dances. However, in this chapter, we use normal RGB streams, i.e. capturing inside the visible spectrum, as inputs to our model. The input can be considered as a time series dataset, where each frame represents an instance of this series. Usually, an LSTM network has better classification performance than a CNN when simple time series are analysed, however, the feature extraction capabilities of a CNN are extremely valuable when analysing visual inputs, as they contain enormous spatio-temporal information. Another difference between LSTM and CNN is that an LSTM network models recurrent and bi-directional capabilities in contrast with the traditional CNN structures. A choreography is a highly temporally dependent video sequence, and therefore, the recurrent model characteristics are significant for dance video modelling.

Thus, we introduce a hybrid deep learning architecture that combines the advantages of an LSTM and a CNN model. In particular, we propose a convolutionally enriched LSTM filter, which operates in RGB video streams, using initially a convolutional layer to extract features for the visual cues, and then feeds these features in an LSTM network. A choreographic primitive usually depends on the past (backward) and future (forward) dancers' steps, resulting in bi-directional (non-causal) relationships. This is due to the fact that dance choreographies consist of a finite number of repeating steps, thus the correct identification of a step provides information about both future and past states.

Moreover, since while non-causal relationships can be covered by the bidirectional capabilities of a bidirectional LSTM network, the classification output is also of use here, as it provides additional knowledge in the classification of both past and future states. To this end, we also include Autoregressive and Moving Average (ARMA) characteristics to the proposed deep learning model. Autoregressive behavior means that the classification output is depended on its own previous value, while moving average properties allow to smooth out

short term classification fluctuations. In this way, the dynamics inherently existing in a dance sequence are addressed.

Another limitation of both LSTM and CNN networks is the assumption of stationarity between the input-output signals. This means that the network weights remain constant throughout the network operation. However, a dance sequence is a highly dynamic sequence. Therefore, network adaptation is needed to fit the current dancer dynamics. For this reason, an adaptive mechanism is introduced to update model response in a way that maximizes overall choreographic modeling performance, addressing different style and gender issues. Finally, we use a foreground estimator exploiting principles of variational inference of Gaussian Mixtures [76]. The purpose of the convolution layer is to transform the high dimensional RGB inputs into low forms of representations, that is the best features for the classification. However, background visual information confuses visual choreographic modeling performance, since it contains data irrelevant to the modeling content. Thus, the convolutional layer operates only foreground data, extracting therefore low dancers' representations which are now less sensitive to motion capturing errors. An overall architecture of the proposed model can be viewed in **Figure 8**.



Figure 8. The proposed adaptive convolutionally enriched LSTM network with AutoRegressive and Moving Average capabilities

4.1. Mathematical Formulation

Let us denote as I(t) an image frame at a time instance t. This frame is processed using the variational inference of Gaussian mixtures of [76]. in order to isolate the background from the foreground. Let us denote as $I_f(t)$ the respective foreground t. The purpose of choreographic modeling is to recognize a set of L different choreographic primitives. For this reason, let us denote as $P_{\omega_i}(t)$ a probability corresponding to one of the L available classes ω_i , i = 1, ..., L. Then, frame I(t) is classified to the class c(t) of maximum probability value

$$c(t) = \underset{\forall \, \omega_i}{\operatorname{argmin}} P_{\omega_i}(t) \tag{1}$$

Usually, there is a non-linear relationship among the raw RGB input data of I(t) and the class probabilities $P_{\omega_i}(t)$. Let us denote as $f(\cdot)$ this non-linear input-output relationship. Therefore, we have that $P_{\omega_i}(t) \sim f(I(t))$. Actually, function $f(\cdot)$ is unknown. One way to approximate $f(\cdot)$ is through the use of feedforward neural networks since it has been proven to be universal approximator [72]. This means that

$$P_{\omega_i}(t) = \boldsymbol{u}^T(t) \cdot \boldsymbol{v}(t) \tag{2a}$$

$$\boldsymbol{u}(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ U_K(t) \end{bmatrix} = \begin{bmatrix} tanh(\boldsymbol{w}_1^T \cdot I(t)) \\ \vdots \\ tanh(\boldsymbol{w}_K^T \cdot I(t)) \end{bmatrix}$$
(2b)

In (2), we have assumed a feedforward neural network of K hidden neurons. v(t) are the weights connecting the outputs of the hidden neurons, denoted as u(t), with the output node, estimating the probability $P_{\omega_i}(t)$. The weights w_i , i = 1,...,K are the ones connecting the input node I(t) with one of the K hidden neurons. Each hidden neuron models the hyperbolic tangent, denoted as *tanh* (see Figure 8).

Autoregressive Moving Average Behavior: The dynamic nature of a dance sequence implies that choreographic modeling depends not only on the current visual observations but also on other backward and forward frames. Moving Average (MA) is often used with input signals to smooth out temporal dependencies. MA is implemented by a Tapped Delay Line (TDL) filter of delaying the input signals of one tap per time. In addition, the output of the dance identification neural model should depend on backward and forward classification outputs mainly due to the dynamics of a dance sequence. Consequently, we introduce an additional AutoRegressive (AR) filter that stimulates the dependence of the classification output on its previous own values. In this way, we ensure a smoothness in the classification output, improving overall choreographic modeling performance.

Long-Range Dependence & Bi-Directional Behavior: A dance sequence follows repeated patterns span on long-time periods, implying a long-range dependence behavior. In addition, choreographic dance modeling follows a bi-directional behavior. For this reason, bi-directional properties are introduced in the fully connected neural network model of (2). Assuming one tap dependence, the following equation is held

$$u_i(t) = \tanh\left(\mathbf{w}_i^T \cdot l(t) + \vec{\mathbf{r}}_i^T \cdot \mathbf{u}(t-1) + \tilde{\mathbf{r}}_i^T \cdot \mathbf{u}(t+1)\right)$$
(3)

Extending (3) to a long-range dependent framework, we conclude to a bi-directional LSTM structure for modeling the $f(\cdot)$. In the LSTM network, the hidden layer is transformed to a memory cell of different processing units, that is *the forget gate, the input node and gate and the output gate* [77].

Convolutionally Enriched LSTMs: The main limitation of the aforementioned structure is that it fails to process efficiently high-dimensional data, such as RGB input signals,

presenting issues related with skeleton errors often occurred by the motion capturing architectures. For this reason, a convolutionally enriched an LSTM structure was used in a way to better process high-dimensional RGB visual signals. In particular, we include a convolution hierarchy after the input layer for transforming the RGB visual signals into low forms of representations. In this way, the convolutional layer is responsible for extracting the skeleton like signal from the raw input data, facing skeleton error related issues.

4.1.1. Network Weight Adaptation

To address the dynamics of a dance sequence, we introduce an adaptive algorithm for dynamic weight modification. Let us now assume that w_a are network weights after adaptation and as w_b before. We now assume that the w_a and w_b are related with a small weight perturbation

$$w_a = w_b + dw$$
 (4)
From (4), it is clear that estimation of the new weights w_a is equivalent to estimate the dw .

It is clear that a dance composes of repeated choreographic patterns. These patterns are periodically appearing through time (perhaps with small variations due to dancer's style). Let us denote as $\pi = \{c_1(t_s), ..., c_N(t_E)\}$ the main choreographic pattern of a dance. A frequency domain approach is adopted for determining this pattern, that is the start and end time instances t_s and t_E [79]. Then, it is clear that the performance of the network should satisfy the main choreographic pattern.

$$y_{w_a}(t) = c_i(t), \forall c_i(t) \in \pi$$
(5)

In Eq. (5), $y_{w_a}(t)$ refers to the network response at time instance t in case that the new (adapted) network weights are used (e.g., w_a). Eq. (5) means that the response of the network within the choreographic pattern should be as close as possible. However, since the network parameters (and consequently the dw) are quite large compared to the number of equations of (5), many solutions satisfy (5). To overcome this difficulty, an additional constraint is introduced; the one that minimizes the norm of dw. Thus,

$$d\widehat{w} = \operatorname{argmin} \|dw\| \text{ subject to} y_{w_a}(t) = c_i(t), \forall c_i(t) \in \pi$$
(6)

Solving Eq. (6), one can estimate the new network weights w_a . In particular, we exploit the assumption of (4). Thus, by applying a first order Taylor series expansion to the LSTM layer

of the proposed neural network model, we can rewrite Eq. (5) as a linear equation system of the form

$$y_{w_a}(t) - y_{w_b}(t) = A \cdot dw \tag{7}$$

In Eq. (7), matrix A depends only from the previous network weight, that is the w_b , and $y_{w_a}(t) - y_{w_b}(t)$ expresses the response difference error of the network of the previous and the adapted network weights over the detected choreographic pattern π .

Variational Inference of Gaussian Mixtures for Foreground Extraction: A variational inference of Gaussian mixtures method is adopted for foreground extraction. The approach presents advantages compared to conventional Gaussian mixtures techniques both in terms of performance and computational complexity. The main difference of a variational inference approach is that the scalar coefficients, regulating the importance of each mixture of Gaussian, is substituted by a probability density function. In other words, each pixel has a probability of belonging to background based on the following probability value

$$P(X) = \sum P(w_i) \cdot N(X, \mu, \sigma)$$
(8)

In Eq. (8), instead of having scalar coefficients to regulate the effect of each Gaussian distribution, we have probability density functions. This means that better approximations can be achieved even for highly dynamic visual environments.

4.2. Performance Evaluation

4.2.1. Dataset Description

Dataset Description & Algorithm Set-up: To evaluate the aforementioned deep learning framework we utilized the latest version of the dataset of choreographic motion capturing of the EU project the TERPSICHORE. The dancers are professionals. Thirty dance sequences have been recorded of different Greek dances [75] presents the description of five dances along with the respective frame choreographic primitive sequence.



Choreographic Steps for Sirtos (3-Beat)

Figure 9. The main choreographic primitives of the Syrtos (3-beat) dance sequence

Figure 9 indicates two choreographic cycles of the dance sequence syrtos (3-beat). As we observed, the choreographic primitives are quite similar with each other, imposing challenges to the classification process. The used algorithms were implemented in Python 3.6 using the Keras and Tensorflow libraries.

Experimental Validation: We have conducted experiments to assess the efficacy of our approach, compared five with other popular classifiers, namely an SVM classifier, a feed forward neural network with 1 hidden layer of 10 neurons, a normal bi-directional LSTM [66] and a CNN [78]. The LSTM classifier is the same as the proposed convolutional LSTM classifier, but without the convolutional layer in the input. The CNN classifier has the same structure of the Convolutional layer of the proposed classifier, followed by on fully connected hidden layer [66]. A comparative Analysis of the performance of these classifiers is presented in **Table 3**.



Figure 10. The effect of background subtraction on the choreographic modelling performance.

Classification Method	Accuracy	Precision	Recall	F1 Score
SVM	60.87%	55.08%	55.35%	55.22%
FNN	52.53%	45.05%	59.23%	51.18%
LSTM [66]	54.89%	47.45%	57.92%	52.16%
CNN [78]	70.57%	70.61%	59.89%	64.81%
Proposed Method	71.35%	71.02%	61.07%	65.67%

Table 3. Comparative Performance Evaluation of the proposed method with other classifiers

Regarding the background subtraction module, Figure 10 illustrates the accuracy of the different classifiers with or without the background subtraction. As is expected, background modelling improves overall choreographic representation. Finally, Figure 11 highlights how the size of the memory window (that is the AR and MA order) affect choreographic modelling performance.





4.3. Conclusions

In this chapter, we have proposed a hybrid CNN-Bidirectional LSTM model for recognition of key choreographic postures in dance sequences. The proposed model combines the multiscale feature extraction process of CNN with the long-term dependency modeling capabilities of bidirectional LSTM networks. ARMA capabilities also included with an adaptive weight modification strategy. The method has been evaluated on RGB sequences depicting real-world sequences of traditional dances and has been shown to outperform other machine learning (including deep learning) approaches in terms of recognition accuracy.

Identification of motion primitives: Space-Time Tensor Based Neural Networks for training under Small Sample Settings

The techniques presented in Chapter 3 and 4 while presenting significant advantages in the way the extract feature and perform the requested micro-action classification tasks, they present one significant drawback, common in most deep learning approaches. That is that deep neural networks are usually extremely complex learning structures with millions of trainable parameters, which makes their training difficult both due to the computational complexity and due to their need of large training sets in order to effectively calculate all these trainable parameters (model weights). Based on this, in this chapter we present the development of a tensor based technique that has the following three advantages. First, we propose an *end-to-end* trainable architecture that unifies the feature and pattern recognition tasks. Second, we exploit tensor algebra tools to significantly reduce the number of the proposed model's trainable parameters making it very robust for small sample setting problems. Last but not least, the proposed approach is a *general* one that can potentially be applied to different problems that employ spatiotemporal data coming from sensor networks.

5.1. Problem Formulation

We consider the problem of human pose classification using 3D skeleton data from Kinect-II. As we will see later, that problem is a specific instance of the more general problem of pattern recognition using information coming from sensor networks. Therefore, in this section, we describe the form of the latter more general problem.

Consider a sensor network that contains C sensors. Each one of the sensors, let's say the c-th sensor, retrieves J measurements (information modalities) at each time instance t, which can be represented by the vector

$$s_{c}(t) = \left[x_{c}^{(1)}(t), \dots, x_{c}^{(j)}(t), \dots, x_{c}^{(j)}(t)\right](1)$$

for $c = 1, \dots, C$. Since each sensor occupies a specific spatial position, the spatial information for the *j*-th information modality captured by the sensor network can be represented by the following vector:

$$s^{(j)}(t) = \left[x_1^{(j)}(t), x_2^{(j)}(t), \dots, x_C^{(j)}(t) \right]$$
(2)

for $j = 1, \dots, J$, while the spatiotemporal information corresponding to a time window *t* to *t* + *T* can be represented by the matrix

$$S^{(j)}(t,t+T) = \left[s^{(j)}(t), \dots, s^{(j)}(t)\right]^T \in \mathbb{R}^{C+T}$$
(3)

The information from all $S^{(j)}(t,t+T)$, $j = 1, \dots, J$ can be aggregated into a tensor object

$$S(t, t+T) = \left[S^{(1)}(t, t+T), \dots, S^{(J)}(t, t+T)\right] (4)$$

in $\mathbb{R}^{C \times T \times J}$. For the sake of clarity, in the following we omit the time index, thus, when we write *S* we refer to a tensor object of the form of (4) for some time instance *t*. Obviously, for a specific time window, the tensor object in (4) encodes the spatiotemporal information for all information modalities and all sensors in a sensor network.

Each tensor S describes a pattern that belongs to a specific class. Let us denote as y the class of that pattern, and assume that we have in our disposal a set D of N pairs of the form:

$$D = \{(S_i, y_i)\}_{i=1}^N \quad (5)$$

The objective of this study is to derive a function for mapping S to y given the set D in (5). This can be seen as a machine learning problem. Let us denote as F the class of functions that can be computed by a learning machine. We want to select the function

$$f *= \arg\min_{f \in F} \sum_{i} l(f(S_i), y_i)$$
(6)

such that $(S_{i_k}y_i) \in D$. In (6) $l(\cdot)$ is a loss function. For classification problems $l(\cdot)$ usually is the cross entropy loss. *Remark 1*: In order to facilitate the solution of problem (6) the learning machine must contain a number of trainable parameters that are comparable to the cardinality

N of set D, and at the same time it should be capable of fully exploiting the spatiotemporal nature of the data.

Remark 2: The problem of human pose recognition using 3D skeleton data from Kinect-II is a special instance of the problem described above. Each skeleton joint can be seen as a sensor, which, at every time instance, measures its x-y-z location. So, in this case *C* equals the number of skeleton joints and *J* in (1) equals 3 (*x*, *y* and *z* positions).



Figure 12. Kinect II skeletal capturing system (vvvv.org/documentation/kinect).

In this chapter, we use 3D skeleton data captured using Kinect-II, along with their annotations, which correspond to the depicted human pose at every time instance. Initially, we process the skeleton data to create tensor objects as in (4) and then use their annotations to create a training set as in (5).

After creating the training set, we design an *end-to-end* trainable neural network, which is able to fully exploit the spatiotemporal nature of the data, and at the same time employs a small number of trainable parameters (compared to the size of the training set). The first layer of the proposed model learns *CSP-like* features from each information modality using inputs in the form of (3). Then, the constructed features from all modalities are fused into a tensor object to compactly represent the spatiotemporal information captured by the sensor network. Finally, the tensor objects are processed by a tensor-based neural network for

producing a mapping from 3D skeleton data to human poses. In the following, we describe each one of the steps presented above in details.

5.2. Data Processing

The Kinect-II sensor identifies and monitors twenty-five skeletal joints at the constant rate of 30 measurements per second, see **Figure 12**. The positions of joints in the 3D space with respect to the Kinect-II device are provided. We utilize the measurements in the form they are captured without employing any tracking technique. A human pose, however, is characterized by the relative positions of the human body parts. For this reason, we represent the position of each joint with respect to the position of the Spine Base joint. This way, the recognition of human poses does not depend on the position of the human with respect to the Kinect-II device.

Specifically, if we denote as $s'_0(t)$ the coordinates of the Spine Base joint and as $s'_c(t)$, $c = 1, \dots, 24$ the coordinates of all other joints, then the coordinates of the joints with respect to the Spine Base joint will be given by

$$s_c(t) = s'_c(t) - s'_0(t), c=1,...,24$$
 (7)

Using the transformed coordinates in (7), we create matrices as in (3) for $j = 1, \dots, 3$ that correspond to x-y-z positions. Those matrices encode the spatiotemporal information for classifying human poses.

At this point, we have to mention that parameter T in (3) is application dependent and affects the recognition results. For this reason, it must be set appropriately. For T = 1, the pose recognition model will not be able to exploit the temporal information and thus it will be more prone to measurements errors, while large values of T may result to a dataset where each datum depicts more than one pose, increasing, this way, the uncertainty in recognition.

5.3. Space-Time Domain Tensor Based Neural Network

The proposed tensor-based neural network consists of three main components; the input layer capable of computing CSP like features, the tensor fusion operation, and the tensor contraction and regression layers that process high-order data in its original multilinear form.

5.3.1. CSP Neural Network Layer

The CSP layer aims to produce highly discriminative features for human pose classification. The design of that layer is motivated by the CSP algorithm , which, for the sake of clarity and completeness, we briefly describe here.

The CSP algorithm originally was developed for binary classification problems. It receives as input zero average signals in the form of (3) along with their labels. Then, its objective is to produce features that increase the separability between two pattern classes. Consider that we have in our disposal *N* samples $\{S_{l,i}\}_{i=1}^{N}$, where l = 1,2 denotes the class of each sample. The CSP algorithm computes the covariance matrix:

$$R_{l,i} = \frac{S_{l,i}, S_{l,i}^{T}}{trace(S_{l,i}, S_{l,i}^{T})}$$
(8)

for each sample, and the average covariance matrix

$$\bar{R}_l = \frac{1}{n_l} \sum_{i=1}^{n_l} R_{l,i}$$
, $l = 1,2$ (9)

for each class, where n_l is the number of samples belonging to class *l*. Then, the CSP filter, *W*, is constructed by using M = 2m, (M < C), eigenvectors corresponding to *m* largest and *m* smallest eigenvalues of $\overline{R_2^{-1}} R_1$. Finally, using *W* each sample is represented by a feature vector:

$$f_{l,i} = \log\left[\frac{var(Y_{l,i}^{1})}{\sum_{j=1}^{M} var(Y_{l,i}^{j})} \dots \frac{var(Y_{l,i}^{M})}{\sum_{j=1}^{M} var(Y_{l,i}^{j})}\right] \in \mathbb{R}^{M}$$
(10)

where $Y_{l,i}^{j}$ stands for the *j*-th row of $WS_{l,i}$. Features $f_{l,i}$ typically are used for as inputs to learning models since they encode the spatiotemporal information of signals $\{S_{l,i}\}_{i=1}^{N}$. Although, theoretically sound, the CSP algorithm presents several drawbacks when applied to real world problems mainly due to the non-stationarity of captured signals. Moreover, it is a feature construction technique that is performed individually, and thus does not permit information flow between feature construction and pattern recognition tasks (see Section I-A2). To overcome those drawbacks, the proposed CSP layer learns *W* during the training of phase model. Trainable matrix *W* projects measurements in R^{M×T} and then features as in (10) are computed from the projected measurements. Additionally, since Kinect-II measurements extract 3D coordinates, we use three parallel CSP layers, one for each coordinate. Therefore, the output of the CSP layer consists of three vectors in R^M.

5.3.2. Tensor Fusion Operation

The fusion module receives as input the feature vectors constructed by the CSP layer and produces a rich and compact representation of the data. Since we do not know in advance the kind of interactions between the elements of the constructed feature vectors, we cannot fuse them using feature averaging or addition. The employed fusion technique is motivated by the work in [11]. The output of the fusion module corresponds to the Kronecker product of the feature vectors produced by the CSP layer. Therefore, after the fusion module each input sample *S*, in the form of (4), is represented by a tensor object in $X \in \mathbb{R}^{M \times M \times M}$. Contrary to [11], we do not reduce the dimensionality of the fused tensor object via decomposition techniques. Instead, we use a tensor-based learning machine capable of processing the fused information in its original multilinear form. The proposed CSP layer and the tensor fusion operation are depicted in **Figure 13**.



Figure 13. The proposed CSP layer and the tensor fusion operation. Parameter N stands for the number of skeleton joints

5.3.3. Tensor Based Neural Network

The employed tensor-based neural network is a fully connected feed forward neural network, its parameter space, however, is compressed [81]. At each layer the weights should satisfy the Tucker decomposition [82]. In particular, the weights W_k at the *k*-th hidden layer are expressed as

$$W_{k} = I_{k} \times_{1} W_{k}^{(1)} \times_{2} W_{k}^{(2)} \dots \times_{J} W_{k}^{(J)}$$
(11)

where I_k is a tensor all elements of which equal one, and the operation "×_j" stands for the mode-*j* product.

The information is propagated through the layers of the tensor-based neural network in a sequence of projections – at each layer the tensor input is projected to another tensor space – and nonlinear transformations. Formally, consider a network with (*K*-1) hidden layers. An input (tensor) sample $X \in \mathbb{R}^{P_1 \times \cdots \times P_j}$ is propagated from the *k*-th layer of the network to the next one via the projection

$$Z_{k+1} = H_k \times_1 (W_{k+1}^{(1)})^T \dots \times_J (W_{k+1}^{(J)})^T$$
(12)

and the nonlinear transformation

$$H_{k+1} = g(Z_{k+1}), \qquad (13)$$

where $g(\cdot)$ is a nonlinear function (e.g. sigmoid) that is applied element-wise on a tensor object. For the input layer $H_0 \equiv X$. The layers that propagate information in the way described above are referred as Tensor Contraction Layers (TCL) [83].

Finally, the output of the (K - 1)-th hidden layer is fed to a Tucker regression model [81], which outputs

$$y_{l} = s \Big(\langle H_{K-1}, (G_{l} \times_{1} W_{K,l}^{(1)}) \dots \times_{J} W_{K,l}^{(J)} \rangle + b_{l} \Big) (14)$$

for the *l*-th class. In (14) the tensor $G_l \in \mathbb{R}^{R_1 \times \cdots \times R_j}$ and R_j is the rank of the Tucker decomposition along mode *j* used in the output layer. The scalar b_l is the bias associated with the *l*-th class, while the subscript *l* indicates that separate sets of parameters are used to model the response for each class. The tensor-based neural network is presented in **Figure 14**.



Figure 14. Propagation of information through the layers of the tensor-based neural network

At this point it should be highlighted that the sequential projections and nonlinear transformations can be seen as a *hierarchical feature construction* process, which aims to capture statistical relations between the elements of the input in order to emphasize discriminative features for the pattern recognition task. Finally, since the weights of the employed tensor-based neural network need to satisfy the decomposition in [84], the total number of trainable parameters is reduced substantially [81]. This reduction acts as a very strong regularizer that shields the network against overfitting [85].

5.4. Performance Evaluation

The dataset of Chapter 3 is used for the evaluation of the tensor based neural network. The dataset consists of four Greek folklore dances performed by three professionals and is publicly available upon request. Each dance performance is described by consecutive frames

and each frame is represented by the spatial coordinates of the twenty-five tracked skeleton joints (see **Figure 12**). The frames of the captured choreographies were manually annotated by dance experts according to the posture they depict. In total seven different postures are depicted. The distribution of annotated samples between different classes for each dance (performer) is depicted in **Table 4**, and apparently the dataset is highly unbalanced. First, we follow the procedure described in 5.2 to transform the coordinates of skeleton joints to a coordinate system in which the origin is the Spine Base joint. Second, we use different values for parameter *T* to create a dataset as in (4). Third, we assign to each sample the annotation of the centered frame, e.g., for T = 15 we assign to the sample the annotation of the 8-th frame.

ID	C1	C2	C3	C4	C5	C6	C7
D1 (P1)	155	201	-	-	44	13	-
D2 (P1)	82	95	42	22	-	-	47
D3 (P1)	122	246	-	-	-	-	82
D3 (P2)	44	268	-	-	-	-	61
D1 (P2)	82	155	-	-	40	85	-
D2 (P2)	82	112	16	32	-	-	44
D5 (P3)	37	98	38	25	-	-	77
D1 (P3)	152	96	-	-	13	16	-
D2 (P3)	33	102	38	25	-	-	77
D3 (P3)	119	130	-	-	-	-	49
Total	908	1503	134	104	97	114	437

Table 4. Distribution of annotated samples between classes for each dance (performer).

For evaluating the performance of our methodology, we randomly shuffle the constructed dataset and follow a 10-fold cross validation scheme. Under that scheme, the performance is evaluated in terms of average classification accuracy and F1 score across the 10 folds. To train our model we used Adam optimizer with learning rate equal to $2.5 \cdot 10^{-4}$. We set the maximum number of training epochs to 300 and employed early stopping criteria to avoid overfitting, which are activated if the accuracy on the validation set is not improved after 20 epochs. The validation set corresponds to 10% of the training set for each fold. Finally, since the problem is unbalanced, we used the weighted cross entropy as the loss function, and the weight for each class corresponds to the inverse of its frequency in the training set.



Figure 15. Average classification accuracy and F1 score of a tensor-based neural network with two TCLs, for T = 7, and for different values of M.



Figure 16. Average classification accuracy and F1 score of a tensor-based neural network with one tensor contraction layer, for M = 24, and for different values of parameter T.

There are three different parameters that affect the performance of the proposed methodology; namely, parameter M, that is the dimension of feature vector constructed by the CSP layer, parameter T, that is the temporal dimension of the samples, and K that is the number of tensor contraction layers employed in the tensor-based neural network architecture.

The effect of parameter M: Parameter M corresponds to the dimension of the features constructed by the CSP layer. For investigating the effect of that parameter on the

performance of the model, we keep fixed the parameter T = 7. Then, we train and test the performance of the proposed model with two tensor contraction layers (TCLs) for different values of M, i.e., M = 12, M = 18, M = 24, and M = 30. The dimension of the tensor contraction and regression layers is presented in the second column of **Table 5**.

The effect of the parameter M is depicted in Figure 15. The best accuracy is achieved for M = 24. The dimension of the features constructed by the CSP layer is directly related to their representation power. Thus, features of higher dimension can better capture the spatial and temporal patterns of skeleton data resulting to more accurate human pose classification. For M = 30, however, the accuracy drops, which might be an indication of over-fitting. Moreover, increasing the value of parameter M increases the total number of trainable parameters of the model. Indicatively, the number of trainable parameters for M equals 12, 18, 24 and 30 is 1335, 1839, 2343, and 2847 respectively.

The effect of parameter T: In contrast to parameter M, parameter T does not affect the number of trainable parameters of the model nor the dimension of the features constructed by the CSP layer due to the variance operator employed in (10). Parameter T indirectly determines the amount of temporal information that is taken into consideration during the construction of the features.

The effect of parameter T on the performance of the model is presented in **Figure 16**. To obtain those results we train a tensor-based neural network with two TCLs and keep the value of parameter M fixed equal to 24. Producing features that encode larger amounts of temporal information results to higher human pose recognition accuracy. Increasing the value of parameter T from 7 to 11 results in a performance improvement more than 10%. Increasing, however, more the value of T results in smaller performance improvements around 2%. This implies that capturing important temporal information for problem at hand more that 11 consecutive frames need to be used.



Figure 17. Average classification accuracy and F1 score of a tensor-based neural network with two TCLs, for M = 24, and for different values of T.

In **Figure 17**, we also compare the performance of the proposed model against a 1D-CNN. First, we concatenated the measurements of different channels to produce input samples for the CNN of dimension $72 \times T$. The CNN performs convolutions along the temporal dimension of the samples, and thus, similarly to the proposed model, it encodes the temporal information within the constructed feature vectors. The employed CNN consists of 3 convolutional layers with 8, 16 and 24 kernels, which are followed by a dense layer with 12 neurons and the output layer. The width of the kernels is (T - 1)/2 for the first two layers and 3 for the third layer. The 1D-CNN and the proposed model perform almost the same. The proposed model, however, employs a significantly smaller number of trainable parameters. Specifically, the proposed model employs 2343 trainable parameters, while the CNN employs 3415, 4631, 5847 and 7063 trainable parameters for T = 7,11,15 and 19 respectively.



Figure 18. Average classification accuracy and F1 score of a tensor-based neural network with different number of tensor contraction layers (parameter K) for M = 24 and T = 11.

3) The effect of parameter K: Parameter K corresponds to the number of TCLs present in the network. Figure 18 presents the effect of the number of TCLs on the performance of the model. To obtain those results we keep parameter M an T fixed and equal to 24 and 11 respectively, and trained four different tensor-based neural networks with 1, 2, 3, and 4 tensor contraction layers. The projections of the employed contraction layers are presented in Table 5. Increasing the number of tensor contraction layers increases the total number of trainable parameters of the model, and thus its learning capacity. Indicatively, the number of trainable parameters for K equals 1, 2, 3, and 4 is 1959, 2343, 2919, and 3783 respectively. That increase, however, does not seem to affect the performance of the model, since the performance improvement from K = 2 to K = 4 is only 1%.

The investigation above suggests that the most important parameter for achieving highly accurate results is parameter M. Indeed, increasing the dimension of the features constructed by the CSP layer from 12 to 24, we achieve a performance improvement of more than 10%. On the contrary, designed deeper architectures does not seem to significantly affect the performance of the model. This might be due to the Tucker decomposition (see (11)), which acts as a very strong regularizer for the model.

	1 TCL	2 TCLs	3 TCLs	4 TCLs
Input	(24×24×24)	(24×24×24)	(24×24×24)	(24×24×24)
Layer1	$(4 \times 4 \times 4)$	$(8 \times 8 \times 8)$	(12×12×12)	(16×16×16)
Layer2	-	$(4 \times 4 \times 4)$	$(8 \times 8 \times 8)$	(12×12×12)
Layer3	-	-	$(4 \times 4 \times 4)$	$(8 \times 8 \times 8)$
Layer4	-	-	-	$(4 \times 4 \times 4)$
TRL	(2×2×2)	(2×2×2)	(2×2×2)	(2×2×2)

Table 5. Projections of tensor objects when they propagated through tensor contraction layers (TCL) and the ranks of the tensor regression layer (TRL).

5.4.1. Performance Evaluation Against State of the Art Methods

In this section we compare the performance of the proposed model against state-of-the-art methods for choreographic modeling. We compare the performance of our model against LSTM and the recently proposed Bayesian Optimized Bidirectional LSTM (BOBi LSTM) [40]. In contrast to the proposed model and the 1D-CNN, the LSTM-based models exploit the order of the data as an additional source of information.

For the performance comparison, we utilize a tensor-based neural network with two TCLs (K = 2), and parameters M and T equal to 24 and 11 respectively. Regarding the LSTM and the BOBi LSTM models, their architectures are the ones presented in section 3 and they use a memory of 10 frames for recognizing human poses. At this point we should emphasize that those models receive as input the kinematic properties of the skeleton joints; i.e., the spatial position as well as the velocity and the acceleration of each joint. In contrast, our method receives as input *solely* the spatial position of the joints. Moreover, the proposed model consists of 2343 trainable parameters. In contrast, the BOBi-LSTM network in [40] was composed by 2 LSTM Layers of 128 cells each and two additional dense layers as the output. This makes the total number of training parameters at 205,674, namely 87 times more than the number of trainable parameters in our approach. This significant reduction favors the efficient parameter estimation especially when small sample setting problems need to be addressed.

Table 6 presents the results of that comparison. The proposed model performs more than

 6% better compared the BOBi LSTM, despite the fact that is uses a simpler input

representation (our method is completely blind to kinematics information of the skeleton joints). Also, **Table 6** presents the performance of the 1D-CNN mentioned above. The 1DCNN performs better than both LSTM models and slightly worse than our proposed model. This implies that models that do not take into consideration the order of the samples are more appropriate for classifying human poses in folklore dances. This is justified by the fact that different dances are composed of different sequences of poses. Therefore, information regarding the order of the samples confuses the model and deteriorates its performance.

Figure 19 presents the confusion matrix for the proposed model. The models performs very well for all classes with the smallest accuracy to be 87% for the second class (cross-legs). 9% of the samples belonging to the second class are misclassified to class 1 (initial pose). This mainly happens due to similarities of the poses belonging to these two classes. For poses that belong to the first and the second classes the dancer faces the camera, and the measurements for all joints (except knees and ankles) are very similar.

The comparison above implies the following. First, the proposed CSP layers can produce highly discriminative features that encode the spatial and the temporal information in the data. Second, employing the tensor fusion operation produces compact yet highly descriptive representations of the input. Finally, tensor contraction and tensor regression layers can efficiently process data in tensor form and produce highly accurate learning models.

	Accuracy (%)	F1 Score (%)
LSTM	84.2%	82.0%
BOBi LSTM	85.4%	80.7%
1D-CNN	91.1%	89.7%
Our Approach	91.6%	90.9%

Table 6. Performance comparison in terms of average classification accuracy and F1 score against LSTM and BOBi LSTM models.

I _								
-	0.89	0.07	0.01	0.00	0.01	0.01	0.02	
2	0.09	0.87	0.01	0.01	0.01	0.00	0.01	- 0.75
ς Ω	0.00	0.01	0.98	0.00	0.00	0.00	0.01	
4	0.01	0.01	0.02	0.96	0.00	0.00	0.00	- 0.50
ŝ	0.06	0.00	0.00	0.00	0.94	0.00	0.00	
g	0.06	0.00	0.02	0.00	0.02	0.91	0.00	- 0.25
~	0.02	0.03	0.01	0.00	0.00	0.00	0.93	- 0.00
	1	2	3	4	5	6	7	0.00

Figure 19. Confusion matrix for T = 11, and M = 24.

5.5. Conclusions

In this chapter we proposed a spatially and temporally aware tensor-based neural network that can efficiently process spatiotemporal data. We evaluated the performance of the proposed model on the problem of human pose recognition using 3D data captured using the Kinect-II sensor. The evaluation results indicate that the proposed model can construct highly discriminative spatiotemporal features and achieve state-of-the-art performance. The problem of recognizing human poses using 3D skeleton data is a specific instance of the more general problem of pattern recognition using information coming from sensor network. Therefore, despite the fact that in this work we consider that specific problem, our model is a general one that can be applied on general pattern recognition problems that employ spatiotemporal data from sensor networks.

Part 2:

Extraction of statistical insights in datasets composed by random sets of actions

6. Supervised Approach: Simulating scenarios in multimodal datasets

In this chapter, we study the use of CNN approaches as presented in previous chapters, in data where the semantic information is not as fine-tuned, but multiple actions are captured simultaneously from both thermal and visual spectrum data modalities, specifically thermal and RGB videos. These datasets are enriched by additional information from other data such as sensors. The learning goal is for the model to be able to ascertain a binary classification paradigm (expected vs unexpected actions).

The need for effective complex representations

Traditional machine learning approaches are sensitive to the features used as input to the detection framework (usually a classifier), therefore appropriate feature selection is crucial. However, extracting adequate features from complex, multi-faceted threats is a very challenging task. This is mainly because of the wide range, variety and heterogeneity of events and their different physical attributes (e.g. pertaining to visual, electromagnetic, mechanical, sensorial information or combination thereof), which cannot be accurately modeled by a common physical law or description. Instead, deep Convolutional Neural Networks inherently compute feature maps extracting complex representations which drive the subsequent classification stage [86] and are especially useful in problems where feature detection and extraction are hard to enact. Furthermore, CNNs exploit strong spatial local correlation by enforcing a local connectivity pattern between neurons of adjacent layers, which can be significant in certain attacks, where locality is a salient attribute.

The need for autoregressive and adaptive learning models

As is often the case, however, there are significant domain-specific factors which even powerful models like CNNs do not inherently take into consideration. In particular, the output of an attack detector should not only depend on external input but also on its classification output history, so as to avoid abrupt spikes in the detection output. Second, an attack detection should often be based on a cumulative behavior over a time period instead of only relying on the current measurable observations, in order to avoid having outliers in the input data trigger erroneous detections. Third, a conventional CNN assumes a stationary input-output relation, whereas this assumption is not valid since a water distribution and monitoring infrastructure environment can be highly dynamic and changing over time. Therefore, an attack detection framework should be based on a non-linear autoregressive and adaptive model that fulfills the above-described conditions.

In this chapter, we explore the intrinsic characteristics and review current methods on detection of attacks pertaining to three information flows: vision-based surveillance, human intrusion detection based on wireless signal reflectance, and cyber-physical attack detection on sensors, actuators or controllers. We then propose a novel framework for multimodal data fusion and adaptive deep learning. The proposed Tapped Delay Line (TDL) CNN model approximate a non-linear Auto-Regressive Moving Average (NARMA) filter. The proposed TDL-CNN classifier achieves an effective feature representation of the heterogeneous input, introduces input- and output memory to the model thus approximating a non-linear autoregressive filter, and incorporates a novel recursive algorithm for online modification the weight parameters of the network to fit the dynamic environmental parameters. An extensive comparative experimental evaluation on real-world data demonstrates the superiority of multimodal vs. unimodal approaches, deep learning vs. "shallow" architectures, as well as autoregressive and adaptive models vs. conventional ones.





6.1. Modelling input data modalities

6.1.1. Visual modality: RGB & thermal camera streams for vision-based detection Computer vision-based surveillance systems are usually based on analysis of RGB video streams. However, the possibility to process information flows from bands beyond the visible spectrum can enhance the performance of intelligent vision systems. In this work, two types of cameras are considered: RGB and thermal. To increase field of view coverage, a network of cameras is used. The raw captured visual data are processed using the YOLO (You look only once) object detection framework. The system models the object detection as a regressive problem by separating the image into spatial bounding boxes and associates to each box a class probability. A convolutional neural network architecture is deployed for performing the object detection task. The model consists of 24 convolutional layers and 2 fully connected layers. For each frame, class object identities are specified per image region (pixel coordinates). In particular, denoting as $o_k(x, y)$ the k-th object identity of the (x, y) pixel, we can form a class label image, say CL(x, y), of the same size as that of the RGB image I(x, y) or the thermal image T(x, y) respectively, so that:

$$CL_i(x, y) \equiv o_{k,i}(x, y), i = \{\text{RGB}, \text{Thermal}\}$$
(1)

where subscript *i* indicates either the RGB or the thermal data. Eq. (2) retains the spatial coherency of the data since the derived class label images are of the same size and spatial consistency with the original raw RGB and thermal image data. For convenience, we resize the RGB and thermal image frames so that they are of equal size, $N \times M$. That is, tensor $\mathbf{x}_{RGB}(n) \in \mathbb{R}^{N \times M}$ represents an image, each pixel of which indicates the object ID that the respective RGB pixel belongs to. Similarly, tensor $\mathbf{x}_{thermal}(n) \in \mathbb{R}^{N \times M}$ represents the class label image of the thermal data. It should be noted that in the case of thermal data, an additional pre-processing stage including background subtraction [87] is carried out. The derived class label image maps, along with the respective confidence scores indicating the reliability in object detection, are the visual (RGB and thermal) modality input, $\mathbf{x}_{RGB}(n) \in \mathbb{R}^{N \times M}$ and $\mathbf{x}_{thermal}(n) \in \mathbb{R}^{N \times M}$ to the multimodal data fusion classifier.

6.1.2. WiFi signal reflection modality for human intrusion detection

Detecting human movement using WiFi commercial off-the-shelf devices can be effected by exploiting Channel State Information (CSI) [88], [35]. CSI models the propagation of a signal from the transmitter to the receiver, supporting many subcarriers due to the Orthogonal Frequency Division Multiplexing (OFDM) principle. The main advantage of CSI data is that they capture physical attributes of the wireless channel, such as scattering, power decay with respect to distance, fading, shadowing and effects of interference [89]. These physical properties are extracted by measuring the amplitude and the phase overall the *K* available subcarriers:

$$H(n) = [H(n, f_1) \ H(n, f_2) \ \cdots \ H(n, f_K)]^T$$
(2)

where $H(n, f_i)$ refers to the amplitude and the phase of the *i*-th subcarrier with central frequency f_i . Therefore, we have that: $H(n, f_i) = |H(n, f_i)|e^{j \angle H(n, f_i)}$.

Usually, H(n) input data contain noise and they are also distorted by the presence of outliers. For this reason, CSI data signals H(n) need to undergo a pre-processing stage. First, outliers are removed using a Hampel identifier [90]. Alternatively, density-based clustering methods such as the DBSCAN algorithm [91] are applied to the raw captured CSI data for outliers' removal. Then, noise is removed by means of wavelet denoising. It should be noted that outlier elimination should precede denoising, since otherwise, outliers may distort the noise removal process. The next stages include normalization, correlation of subcarriers and eigenvector processing of the signals (**Figure 21**).

The pre-processed CSI data are analysed using a linear Support Vector Machine (SVM) classifier in order to detect human intrusions in a scene, which constitutes the output of unimodal detection based on WiFi signal reflectance. These classification IDs, say $C_{WiFi}(n)$, will also be used as input to the proposed fused deep learning classifier for cyber-physical attack detection. Therefore, tensor $\mathbf{x}_{wifi}(n)$, pertaining to the WiFi signal reflection for human intrusion detection modality, is composed of:



$$\boldsymbol{x}_{wifi}(n) = [H(n) \ C_{WiFi}(n)]^T$$
(3)

Figure 21. Schematic overview of human presence detection mechanism from WiFi reflection signals.

To retain spatial coherency in line with the visual input data, tensor $\mathbf{x}_{wifi}(n)$ is expanded over the $R^{N \times M}$ grid, forming an additional input channel to the multimodal detection framework.

6.1.3. ICS sensing modality: PLC and SCADA data for cyber-physical attack analysis Interconnected sensors and controlled devices of critical infrastructures, like water utilities, have been primarily designed for industrial process control. They provide valuable information about the smooth operation of the infrastructure, and can be utilized for security and protection purposes in an appropriately designed holistic threat detection framework. Extraction of appropriate features for ICS measurable data (e.g. sensors, PLC and SCADA indications) monitored by the operator is not straightforward, since there is no direct physical interpretation of cyber threats with sensorial patterns appearing in the monitoring signals [92]. Moreover, there are several types of different attacks with different "signatures", which makes them difficult to model holistically. To address these challenges, the proposed TDL-CNN multimodal deep learning model is allowed to find the most appropriate features in a way that classification performance for detecting cyber-physical attacks on specific application scenarios is maximized.

Therefore, tensor $\mathbf{x}_{ICS \ sensing}(n)$ comprises a set of measurable sensorial data obtained from ICS of a water utility infrastructure. As in the previous case, the data are pre-processed so as to eliminate outliers and noise, using DBSCAN algorithm and a low-pass filter respectively. In this work, we measure the flows of two water pumps, the suction pressure and the discharge pressure for a real-world water utility. The measurements are acquired at 30 sec intervals. Again, to retain spatial coherency as for the previous cases, we expand tensor $\mathbf{x}_{sensing}(n)$ over the $R^{N \times M}$ grid, forming again an additional input channel.

6.1.4. Multimodal data fusion from visual, WiFi reflection and ICS sensing input channels

Unimodal approaches based on solely one of the above types of information are bound to have limitations as regards the range of threat types that they can detect. Critical infrastructures today may face increasingly sophisticated multi-faceted attacks, protection from which unavoidably requires a holistic approach that intelligently combines different channels of information. In this context, the adaptive deep learning model proposed is driven by a fused multimodal data tensor. Therefore, the multimodal input tensor data x(n) can be derived as:

$$\boldsymbol{x}(n) = \begin{bmatrix} \boldsymbol{x}_{RGB}(n) \ \boldsymbol{x}_{thermal}(n) \ \boldsymbol{x}_{wifi}(n) \ \boldsymbol{x}_{ICS \ sensing}(n) \end{bmatrix}^{T} \quad (4)$$

where $\mathbf{x}_{RGB}(n)$ is the data tensor pertaining to RGB visual signals, $\mathbf{x}_{thermal}(n)$ the respective data tensor of the thermal component, $\mathbf{x}_{wifi}(n)$ the data tensor pertaining to the WiFi reflection signal and, finally, $\mathbf{x}_{ICS \ sensing}(n)$ the data tensor of the ICS sensing modality.

6.2. The proposed adaptive deep learning model for cyberphysical event detection

6.2.1. Tapped Delay Line Convolutional Neural Network (TDL-CNN)

Let us denote as $y(n) = [p_{\omega_i} \cdots p_{\omega_L}]^T$ an $L \times 1$ vector that contains probabilities p_{ω_i} for attacks ω_i (out of L possible ones) occurring in the water utility infrastructure at time instance n. Classes ω_i may correspond, e.g., to cyber threat, physical intrusion, a combined attack detection, or a normal functional situation; such a scheme is also adopted in the experimental evaluation. Let us now assume that there is a non-linear function that relates probabilities p_{ω_i} with some measurable input observations x(n) that describe the status of the critical water infrastructure at time instance n. To calculate probabilities p_{ω_i} we need to take into account several previous observations over a time window consisting, say, of qprevious time instances. That is, vector y(n) depends on q previous samples x(n - j), j=0, \dots , q-1. Furthermore, the classification also depends non-linearly on its own previous values, thus resulting in a *non-linear autoregressive-moving average framework*. Therefore, the classification output y(n) can be modelled with a non-linear vector-valued relationship $g(\cdot)$:

$$y(n) = g(x(n-1), \dots, x(n-q), y(n-1), \dots, y(n-p)) + e(n)$$
(5)

where, p, q express the order of the model over the previous q measurable observations and previous p classification values. Additionally, vector e(n) is an independent and identically distributed (i.i.d.) error. The main difficulties in Eq. (5) are that: (i) non-linear relationship $g(\cdot)$ is actually unknown, and (ii) input observations x(n) should be properly selected so that we can suitably divide the attack classification space in a way to maximize attack classification performance.

To address the first fact, machine learning methods can be applied to approximate $g(\cdot)$ in a way that minimizes error e(n). Eq. (5) actually models a Non-linear Autoregressive Moving Average (NARMA) filter. In particular, a feedforward neural network (FNN) with a tapped delay line (TDL) input filter can simulate the behavior of a NARMA(p,q), while a recursive implementation of such a model has been proposed in [72]. However, such a TDL-FNN model fails to address the challenge of effective feature selection in a high-dimensional space and a complex heterogeneous environment. In this context, Convolutional Neural Networks (CNNs) have demonstrated excellent representational capabilities in feature selection [86].

The proposed TDL-CNN model combines the representational power of CNNs with the autoregressive nature of TDL. A TDL-CNN selects the optimal features for classification through an approximation of a series of convolutional filters, while also modeling the unknown vector-valued relationship $g(\cdot)$ of Eq. (5). To this end, we expand the architecture of a CNN by (i) adding a TDL input layer which acts as a spatiotemporal moving average of the multiple modality input channels, and (ii) feeding back the classification output as additional input to the network over a time window. A block diagram of the proposed architecture is shown in **Figure 22**.



Figure 22. Architecture of the proposed TDL-CNN.

Tapped Delay Line Layer: The purpose of this layer is to appropriately organize the external input data x(n) as well as to feed back the previous classification outputs. It consists of two terms: The first term models the moving average component by delaying the external input signals x(n) for q discrete previous times. The second term simulates the autoregressive component by delaying the output of y(n) over a time window of p previous discrete times. The TDL is a non-linear dynamic model, employed to endow the network with an autoregressive character. Past classification results influence current and future outputs to an extent, as temporal dependencies do occur. Therefore, the TDL layer helps take into consideration previous classification results, thus decreasing spikes in the output behavior.

Convolutional Layer: The purpose of this layer is to apply convolutional transformations on the input data in a way as to maximize classification performance. A set of parameterizable filters (e.g., learnable kernels) is convolved with the input data selecting appropriate features and estimating kernel parameters, so that performance error on a ground truth training set is minimized. The *L* feature maps, say $f_1, f_2, ..., f_L$, optimally selected by the convolutional layer will be used as input to the final classification layer.

Classification Layer: The Classification Layer receives the transformed representations from the convolutional layer as input, i.e. feature maps $f_1, f_2, ..., f_L$, and triggers the final (supervised) attack predictions. Normally, feature maps f_i are tensors of a high dimensional grid. The first dimensions express the spatial attributes of the scene, in 2D or 3D space, while the rest refer to the different modalities of the input data. In the following, to simplify the notation, we assume, without loss of generality, that feature maps f_i are scalars. Extension to tensors can be done by exploiting tensor algebra properties and appropriate modification of the inner product operators.

Let us now assume that the classification layer consists of one hidden layer of r neurons. Each neuron stimulates a non-linear operation, modeled by an activation function $\varphi(x)$. Usually, the sigmoid function is used. Let us denote as $w_{i,j}$ the weights that connect the *i*-th feature map, expressed by f_i , with the *j*-th hidden neuron of the classification layer. Then, the output of this neuron will be $u_j = \varphi(w_j^T \cdot f)$, where f is the aggregate feature map including all features f_i and w_j the aggregate weights for the *j*-th hidden neuron, i.e., all weights connecting all feature maps with the *j*-th hidden neuron. Then, output will be given as:

$$y_w(n) = \varphi(\mathbf{v}^T \cdot \mathbf{u}) \equiv \varphi(z_w(n)) \tag{6}$$

where *u* includes all outputs u_j of the *r* hidden neurons and **v** the aggregate *r* weights connecting the *r* hidden neurons of the classification layer with the output neuron. In Eq. (6), $z_w(n)$ expresses the input of the final output neuron before applying the activation function $\varphi(\cdot)$. Here we have assumed that, without loss of generality, the classification output consists of one neuron. Extending to multiple output neurons is simple. In Eq. (6) we have added the dependence of the classification output $y_w(n)$ on network weights *w*, estimated by a training process.
6.2.2. Adaptive TDL-CNN

The main limitation of the TDL-CNN is that it assumes a stationary stochastic non-linear relationship of the input data with the classification output. However, this cannot be the case in real-world application scenarios, as in a modern critical infrastructure monitoring setting, due to the dynamic nature and complexity of the system, and the elaboration of potential attacks. Therefore, adaptation strategies are required to recursively update the model's behavior through appropriate weight modification to fit the changing environmental conditions.

Let us denote as $w^{(1)}$ all the weights of the classification layer before the adaptation, and $w^{(2)}$ the respective weights after the adaptation. Then, we assume that these weights are related via a small perturbation factor dw: $w^{(2)} = w^{(1)} + dw$. It is clear that estimation of the new weights $w^{(2)}$ is equivalent with the estimation of dw. To calculate dw, two complementary types of constraints are considered: *discriminative and generative constraints*.

The discriminative constraints model the current statistics of the input-output relationship that fit current environmental conditions. In particular, we assume that a training set $S_c = \{(x_i(n), t_i(n))\}$ includes pairs of input-target relationships at a time instance *n*. Input data $x_i(n)$ express fused information from multiple modalities, while targets t_i are supervised (desired) outputs, provided by water utility experts. Then,

$$y_{w^{(2)}}(x_i, n+1) \approx t_i(n), \ \forall (x_i, t_i) \in S_c$$
(8a)

or

$$z_{w^{(2)}}(x_i, n+1) = \varphi^{-1}(t_i(n)) \equiv d_i, \ \forall (x_i, t_i) \in S_c.$$
(8b)

Eq. (8) means that the small weight perturbation is estimated so that the current collected data (by set S_c) are trusted as much as possible. By applying perturbation theory and particularly a first order Taylor series expansion on the last classification layer of the network we can conclude to a linear relationship for dw:

$$c_i(n+1) = A_i \cdot dw \tag{9}$$

where $c_i(n + 1) \equiv z_{w^{(2)}}(x_i, n + 1) - z_{w^{(1)}}(x_i, n + 1)$ is the classification difference before and after the adaptation, A_i is only related with the previous network weights $w^{(1)}$. The generative constraints model the effect of the already obtained knowledge on weight updating, yielding to stable adaptation solutions. Previous knowledge is modelled by a set S_p of the same structure as S_c . The effect of small perturbation dw is expressed by applying sensitivity analysis. Taking into account both constraints, we conclude to:

 $E = \frac{1}{2}dw^T \cdot J_{s_p}^T \cdot J_{s_p} \cdot dw \ \forall i \in S_p$ (10a) subject to $c_i(n+1) = A_i \cdot dw, \forall i \in S_c$ (10b) where J_{s_p} expresses the Jacobian matrix over set S_p . The aforementioned constraint minimization consists of a convex term (see Eq. (10a)) subject to a linear constraint (Eq. (10b)). Iterative methods are applied for solving (10), such as the reduced gradient method [72].

6.3. Experimental evaluation

6.3.1. Experiment setup

The dataset used to evaluate and validate the proposed methods has been captured as part of the EU Horizon 2020 STOP-IT project (<u>https://stop-it-project.eu/</u>), a research initiative that addresses the protection of critical water infrastructure and that includes as consortium members eight (8) water utilities from Spain, Israel, Germany, and Norway that are responsible for providing water distribution services in more than ten million citizens in total.

The dataset consists of RGB and thermal camera streams, data from WiFi reflectance based detection and ICS data. In particular, the RGB data were captured using using OB-500Ae cameras with a 1280×720 pixels resolution and a 30 fps framerate. The thermal data were captured using Workswell InfraRed Camera 640 (WIC) with a 640×512 pixels resolution and a 30 fps framerate. To acquire WiFi signal reflection data, two WiFi devices were used. The WiFi router (TP-Link N300 TL-WR841N) implements the 802.11n standard, used to retrieve the CSI information. For the receiver, Intel's 5300 NIC was plugged in to a standard laptop. This setting allows data capturing at 10 sec intervals. Finally, the ICS sensing data consist of information from water infrastructure SCADA systems and include the pressure of two pumps, suction pressure, discharge pressure, and the water level from a water tank. The computer used for all training and testing was an Intel® CoreTM i7-6700 CPU@ 4000

GHz CPU with 16GB of RAM and an NVIDIA GeForce GTX 1070 with 8GB DDR5 memory. The deep learning models also used the CUDA 9.2 Toolkit.

Data are labeled based on pre-determined scenarios co-defined by end users, i.e. water utilities, that designate: normal behavior, cyber attacks (on ICS sensors), physical intrusions (including tracking of suspicious movements in secured areas from the RGB and thermal cameras, as well as intrusions not captured by cameras but detected from WiFi reflection) and a combination of both cyber and physical attacks (notated in the dataset as cyber-physical attacks). All data are normalized so as to be in the same range, i.e. from 0 to 1. The dataset consists of 5 days of data, including individual attacks per modality, so that the dataset is sufficiently representative of attack patterns. The ICS modality includes 24 instances of hour-long attacks. The RGB, thermal and WiFi modalities, were all captured simultaneously. They include 20 different instances of attacks, spanning in duration from 2 to 20 minutes of consecutive suspicious behavior.

Regarding the details of the proposed TDL-CNN model, it is implemented through (i) the TDL input layer, (ii) the convolutional/pooling layers and (iii) the classification layer. The input layer receives the current data (RGB, thermal, ICS, and WiFi) along with tapped delay responses over previous times. It includes three Convolutional/Pooling layers with a convolutional, ReLU and a Max pooling component. Finally, the classification layer consists of one fully connected hidden layer and one output layer. The first convolutional layer consists of 32 filters with a filter size of 5x5x4 (three RGB channel plus one thermal channel; the remaining modalities are added as additional rows over all four channels), the second again of 32 filters of size 5x5x32 (since 32 filter kernels are produced by the first convolutional, ReLU and Max pooling component) while the third of 64 filters of size 5x5x32. The stride for the convolution for all layers is 1x1 while the polling stride is 2x2. Finally, the classification layer consists of 64 hidden neurons and 4 output neurons. The input size of the TDL-CNN is 640x512x3 for the RGB, one additional 640x512x1 for the Thermal and 12 additional rows of data for the ICS and WiFi modality). The feature map produced by the convolutional/pooling layer is 576.

6.3.2. Results

We have conducted extensive experiments to evaluate the efficacy of the proposed approach and showcase the contribution of each one of its core components, i.e. fusion from multiple data modalities, deep learning, and finally autoregressive and adaptive capabilities.

Regarding the significance of data modalities utilised for attack detection, Table 7 shows the classification performance in cases where only one information modality is taken into consideration: (i) visual (RGB and thermal), (ii) WiFi signal reflection, or (iii) ICS sensing modality. Four different classifiers were used: a linear kernel SVM, a non-linear Radial Basis Function (RBF) kernel SVM, a Feedforward Neural Network (FNN1) with 1 hidden layer of 10 neurons, and another FNN2 with 2 hidden layers of 10 neurons/layer. Classification performance is measured through five objective metrics, namely Precision, Recall, False Positive Rate (FPR), Accuracy and F1-Score. As is observed, the classification performance is low when data from a single modality are used as input. We also observe that the classification performance on ICS sensing and WiFi signal reflection modality is almost the same over all classifiers. This is mainly due to the fact that simple data taken from interconnected sensors of a water utility do not suffice to lead to detection of complex unusual activity and combined cyber-physical attacks. In all results of Table I, we have assumed that the autoregressive and moving average window (p, q) of past time instances is 100 frames long (henceforth referred to as "long memory" case).

Table 8 depicts attack detection performance in case that fused data across multiple modalities are used as input (again, for the "long memory" case). In this case, apart from the "shallow" machine learning models mentioned above, deep learning schemes are additionally employed. In particular, we scrutinize the effectiveness of: a Long Short-Term Memory (LSTM) deep recurrent neural network, a conventional Convolutional Neural Network (CNN) and the proposed Tapped Delay Line CNN (TDL-CNN), as well as the adaptive versions of CNN and TDL-CNN. As is observed, data fusion from all three modalities significantly improves classification rates even in the case of shallow classifiers. Moreover, performance rates improve significantly when deep learning schemes are utilized, which highlights the representational power of the models and their suitability for the discussed critical infrastructure monitoring application. We also notice that the proposed

TDL-CNN, i.e., a CNN network with autoregressive-moving average properties, yields the best performance in terms of all metrics (barring its adaptive version, which will be elaborated on later).

Table 8 shows the execution times (per 100 frames) for the multimodal configurations. As can be observed, the proposed adaptive TDL-CNN's execution time (1.36452 sec per 100 frames) is only a little higher than that of the plain CNN (0.91556 sec) and the shallow models, although its respective classification performance is higher. In all cases, the processing time remains lower than 25 frames/sec, i.e., less than 40 msec per image frame. It is also observed that the proposed adaptivity mechanism minimally increases the execution time. Training of deep learning is of course computationally more demanding compared to conventional methods (it takes approximately 1.2-1.8 hours to train shallow models, as opposed to 7-7.5 hours for LSTM and CNN frameworks, and around 15 hours for TDL-CNN). However, the training process is an offline process that only takes place once; then the adaptability of the proposed self-configurable scheme readjusts the network parameters to better fit new behavior instances, thus obviating the need for a new retraining phase.

Classification Method	Precision	Recall	FPR	Accuracy	F1 Score
Visual Modality					
SVM-Linear	40.02%	27.42%	29.57%	52.43%	32.55%
SVM-RBF	25.98%	26.34%	54.00%	37.78%	26.16%
FNN1	35.04%	41.35%	56.16%	43.38%	37.93%
FNN2	40.21%	59.98%	64.16%	45.94%	48.14%
WiFi Signal Reflection Modality					
SVM-Linear	22.06%	24.59%	62.50%	32.10%	23.25%
SVM-RBF	22.42%	24.68%	61.45%	32.74%	23.50%
FNN1	22.03%	26.34%	67.09%	30.16%	23.99%
FNN2	22.84%	26.21%	63.69%	32.08%	24.41%
ICS Sensing Modality					
SVM-Linear	29.37%	31.30%	54.15%	39.76%	30.31%
SVM-RBF	29.37%	31.30%	54.15%	39.77%	30.31%
FNN1	29.37%	31.30%	54.15%	39.76%	30.31%
FNN2	29.37%	31.30%	54.15%	39.76%	30.31%

 Table 7. Classification performance metrics for experiments using a single data modality (visual, WiFi signal reflection, ICS sensing). Four different classification methods have been examined.

Classification Method	Precision	Recall	FPR	Accuracy	F1 Score	Execution
						Time (per 100
						frames)
"Shallow" Models						
SVM-Linear	59.04%	62.71%	31.30%	66.19%	60.82%	0.72897 sec
SVM-RBF	43.19%	50.14%	47.45%	51.54%	46.40%	0.75852 sec
FNN1	49.08%	64.68%	48.29%	57.14%	55.81%	0.89652 sec
FNN2	51.49%	63.94%	43.35%	59.70%	57.04%	0.92356 sec
Deep Models						
LSTM	70.38%	62.63%	18.97%	73.33%	66.28%	0.90547 sec
CNN	74.20%	71.20%	17.79%	77.60%	72.68%	0.91556 sec
Adaptive CNN	75.14%	74.72%	17.79%	79.08%	74.93%	0.95667 sec
TDL-CNN	84.45%	78.77%	10.44%	85.04%	81.51%	1.36448 sec
Adaptive TDL-CNN	85.41%	84.92%	10.44%	87.62%	85.16%	1.36452 sec

Table 8. Classification performance metrics and execution times (per 100 frames) for experiments using fusion of all modalities (visual, WiFi signal reflection, and ICS sensing).

In the sequel, the effect of the autoregressive – moving average property is examined for the multimodal fusion experimental setting. **Figure 23**(a) depicts the respective effect in case that shallow learning classifiers are exploited, whereas **Figure 23** (b) illustrates the same results when deep learning schemes are employed. For all cases, as the length of the memory window increases, better performance rates are noticed, but a saturation in the improvement is also encountered. Deep machine learning classifiers yield better performance than the conventional shallow ones as is also shown from **Figure 23** (b) where the best performing shallow classifier (FNN2) is overlaid with the deep learning schemes.

The same autoregressive – moving average performance is noticed for the unimodal visual case (see **Figure 24**(a)) but reaching far lower classification rates than the multimodal case. However, in the case of unimodal WiFi signal reflection and ICS sensing data, the autoregressive – moving average effect is minimal and the results are constant regardless of the memory window length used. This is clearly shown in **Figure 24** (b), in which we compare the effect of the memory window length on the F1-score for each one of the three unimodal settings in the case of SVM-Linear and FNN2 (feedforward neural network with 2 hidden layers and 10 neurons/layer).

Finally, **Figure 25** shows how the proposed adaptive scheme can further improve the performance of both the conventional CNN and, more importantly, the proposed TDL-CNN model, which attains an overall accuracy of 87.62% and a F1-score of 85.16%. It is clear that using the adaptation, a small but consistent improvement in all performance metrics is noticed; this is explained by the fact that the classifier can automatically adjust to the changing dynamics of the environmental and application-specific conditions, let alone requiring a very small number of samples for the readjustment process.



Figure 23. The effect of autoregressive – moving average behavior on the classification performance (F1-score) in the case of multimodal data fusion using (a) shallow learning classifiers and (b) deep learning ones. Short memory corresponds to considering 30 previous frames, while long memory corresponds to 100 previous frames.



Figure 24. The effect of autoregressive – moving average behavior on the classification performance (F1-score) in case that (a) data from only the visual modality are used, and (b) data from only WiFi signal reflection or ICS sensing compared to the respective behavior on the unimodal visual case.



Figure 25. Performance metrics for CNN and the proposed TDL-CNN, as well as their adaptive versions. Applying the adaptive scheme improves the classification performance in terms of all metrics examined.

Overall, the successful performance of the proposed model can be explained by a combination of factors. Intertwining different information modalities offers increased insight into the complex multi-faceted nature of water infrastructure attacks. These can be successfully modeled by means of deep learning models, due to the great generalization (as opposed to memorization) capability of the latter [93]. Furthermore, the autoregressive property of the proposed TDL-CNN plays a significant role in "smoothening", i.e. removing spikes from the output. Finally, the adaptive mechanism endows the model with a reconfigurable behavior that allows self-adjustment to dynamic settings and thus mitigation of misclassification error.

6.4. Conclusions

In this chapter, we highlighted the significance of using multiple data modalities, i.e. RGB, thermal, WiFi signal reflection, and ICS sensor data, as a driver for a cyber- and physical attack detection. To address the challenges involved, we proposed an extension of the CNN model, the Tapped Delay Line Convolutional Neural Network (TDL-CNN), which combines the representational power of deep learning with autoregressive and moving-average attributes of a NARMA filter. An additional adaptive version of the TDL-CNN was presented, which allows the model to better adapt to dynamic attack characteristics.

The proposed methods were experimentally evaluated using a dataset captured in the context the EU H2020 STOP-IT project. The results show that the use of multimodal data fusion leads to significantly better attack detection rates compared to unimodal approaches; the same goes for deep CNNs compared to "shallow" models. Finally, the results indicate that the autoregressive and adaptive attributes of the proposed multimodal deep model provide clear added value in terms of the performance rates attained in cyber and physical attack detection.

7. Unsupervised Approach: Fall detection in optical and thermal datasets

All the aforementioned techniques are paradigms of supervised learning, where an annotated dataset is present and helps the training of the deep learning models. However, all those approaches, even powerful extended deep convolutional networks lack in the area of generalization. This means that the overall model fails when we "change the scene" of application, and for each installation, a necessary annotation needs to take place before the training. To this end, unsupervised approaches, while not offering the same degree of granularity in terms of action recognition, as the resulting semantic models are cruder than the fine-tune micro-action identification that can take place with supervised approaches, they however provide valuable insights in terms of modeling the statistical distribution of various actions composing this dataset.

This means, that the detection of actions that are not expected to be part of the composition of the data that are under analysis, can take place by using deep learning techniques to essentially model the normality within the data. Autoencoder approaches, have been proven quite valuable in this area. Autoencoders are models that learn to extract representative features (encoding part) from the input data. These features are selected during the training process, with the overall learning goal being to be able to use these features to re-extract the high dimensional input signals (decoding part). The advantage of these approaches is that we can train the autoencoder using only examples from a normal situation. Then by simply monitoring the reconstruction error, in an already trained model, we can deduce the presence of outlier events, simply by the fact that the autoencoder failed to extract a representation of them.

The first application scenario for the testing of such an approach presented here is a fall detection scenario, specifically, a man-overboard event. A man overboard is an emergency incident, where a crew member or passenger of a maritime vessel has fallen off-vessel in the sea. These types of accidents are more often in passenger ships, where there is presence of a large number of untrained individuals. It is estimated that 22 people fall off a

cruise ship annually [94]. Moreover, these incidents have high mortality rates, as almost 79% of the victims do not survive or are considered missing [94]. The cause of such high motrality rates is the low speed of detection and retrieval. After an hour in water at 4.4°C, body temperature drops to 30 °C [95]. Thus, it is a critical event that demands immediate handling as time plays an important role and because the overboard casualty is exposed to various security risks, such as drowning at sea, hypothermia, injuries and rough sea. It is noted that the problem lies in the lack of timely and critical information, such as the accurate confirmation of the event as well as its exact time and position of the occurrence. The proposed framework (see Figure 26) is based on a spatiotemporal convolutional autoencoder, which is trained on RGB video sequences that simulate man overboard scenarios. We train our network on the normal situation in order to learn efficient data encodings by ignoring signal noise and then use its reconstruction error to detect man overboard as an abnormal event during the test process. In parallel, we utilize multiple image proper-ties to enhance the identification capabilities of the proposed architecture. To the best of our knowledge, man overboard identification has not been addressed as an anomaly detection task utilizing unsupervised deep learning techniques.





The presented system using only RGB video streams to identify overboard falls. However, the simple use of raw RGB frames is not sufficient for an efficient detection. To extract additional data from the visual modality we furtherly analysed the camera streams to extract specific visual properties, i.e. representative vectors. To this end, the visual modality is analyzed to extract the actual frame (appearance), the gradient of the frame using a short memory window of 10 frames (movement vector), the objectness of the current frame (saliency vector). The Appearance Property consists of the actual frame capturing. The Motion Property captures the movement of objects by taking as input the gradient of the frame. Finally, the Saliency Property reflects how likely a window of the frame covers an object of any category. This property creates a saliency map with the same size as the frame that covers all objects in an image in a category independent manner.

Each image property was fed into an individual spatiotemporal autoencoder. Autoencoders are a type of Neural Network that manage to learn efficient data encodings by training the network to ignore signal noise. Their usefulness comes from the fact that they are trained in an unsupervised manner. They are essentially composed from two main components that are trained in parallel. The dimensionality reduction component aims at extracting an efficient encoding of the input signal, while the reconstruction side tries to generate from the reduced encoding a representation as close as possible to the original input. To identify the abnormalities, the reconstruction error of each autoencoder was monitored, and when the error was bigger than a predefined threshold, an alert was raised. The selection of the threshold took place during the training, to identify the exact value that maximized detection performance.

The autoencoders used for each image property had the structure presented in **Figure 27**. Each RGB frame for the appearance vector was reduced to a grayscale image with a resolution of 227x227x1. A 10 frame batch was used for the analysis. Each autoencoder had the structure presented in **Figure 28**.



Figure 29. Individual Autoencoder Structure

7.1. Stacked Convolutional Autoencoders for Feature Extraction

The property cubes P_k generated by the aforementioned property operators are usually sparse tensors containing redundant information. For this reason, a stacked convolutional autoencoder [31] has been utilized for compressing the tensors P_k , acting as an intra-property compression scheme. In this chapter, we chose convolutional autoencoders, instead of the traditional neuron-based models, since convolutional filtering is more suitable for processing and analysis of multidimensional imaging signals.

A convolutional autoencoder is trained so that its target output coincides with the autoencoder input itself, resulting, therefore, in an unsupervised learning paradigm, since labelled (annotated) data are not required during the learning process. It has, in general, two main parts; the encoder which is responsible for compressing the image data through learning and the decoder with the main purpose of best reconstructing the input signal from the compressed, encoded data.



Figure 30. Structure of the encoding part of a stacked convolutional autoencoder



Figure 31. Structure of the Convolutional Kernel Operator

Let us denote, in the following, as *L* the number of encoding hierarchies of the model. As in the traditional autoencoders, where each hidden layer is constructed by a number of neurons, processing the input signal through an inner product operator, the encoding layer of a convolutional autoencoders is constructed by a number of convolutional kernels. A convolutional kernel of an encoding layer performs three main types of operations; a convolution, a function activation and a max-pooling. **Figure 32** presents the architecture of the encoding part of a stacked convolutional autoencoder, while **Figure 33** the main operators of a convolutional filter, which is the heart of the autoencoder.

First, the input signal is convoluted with a filter kernel, defined by a set of weights $w_i^{(l)}$. In this notation, $w_i^{(l)}$ refers to the *i*-th convolutional filter of the *l*-th encoding layer.

Then, the convoluted image is fed to a non-linear activation function, performing value adjustment through pixel-based processing.

Finally, a max-pooling operator is considered which is responsible for the compression (down-sampling) of the input data. Therefore, the output of a convolutional kernel is

$$g_{i}^{(l)} = \sigma(w_{i}^{(l)} * C(L-1))$$

$$c_{i}^{(l)} = \max_{pooling} \left(g_{i}^{(l)}\right), l = 1, 2 \dots, L \quad (1)$$

In Eq.(1), the operator '*' corresponds to the convolution between the input signal C(l-1) and the filter $w_i^{(l)}$. The $\sigma(\cdot)$ refers to the non-linear activation function. Example of $\sigma(\cdot)$ are the sigmoid, the hyperbolic tangent, and the rectified linear unit (ReLU) functions. Tensor $c_i^{(l)}$ refers to the final output of the *i*-th kernel at the *l*-th encoding layer. Finally, tensor C(l-1) refers to the input signal of the convolutional kernel.

Actually, the tensor $c_i^{(l)}$ is a codeword or a representation of the input signal P_k at the *l*-th encoding layer derived by the convolutional kernel $w_i^{(l)}$. Gathering all these individual codewords $c_i^{(l)}$, together, we form a codebook representation C(l) of the input signal P_k at the *l*-th hierarchy:

$$C(l) = \{c_1^{(l)}, c_2^{(l)}, \dots, c_{Q_l}^{(l)}(2)\}$$

It is clear that $C(l = 0) \equiv P_k$ since at this layer no compression is encountered. In Eq.(2), Q_l is a scalar denoting the number of convolutional filters at the *l*-th encoding layer. A codebook C(l) is propagated at the next encoding layer feeding as input the convolutional kernels of the next hierarchy. Therefore, a hierarchy of codebooks are created $C(1), C(2), \dots, C(L)$. The convolutional kernels of the network, which are used to compute the codebooks C(l) of Eq.(2) are estimated through a learning process so that the codewords are optimally reconstruct the input signals. That is,

$$E_{c} = ||P_{k} - f(C(l))||_{2}$$
(3)

where $|| \cdot ||_2$ represents the mean square error and $f(\cdot)$ is a non-linear function of the decoder part of the autoencoder modelling through inverse convolutional operators of the encoder. Since the convolutional autoencoder has L encoding layers, the codebook used for representing a property P_k is the one derived from the last encoding layer $C_k(L)$.

7.2. Evaluation

7.2.1. Dataset Description

To train and evaluate the proposed methodology, a mock man-overboard event was conducted that concerned the fall of a human-sized dummy from the balcony of a high-rise building. In particular, the human dummy (see **Figure 34**), weighting 30 Kg, was thrown from an approximate height of 20 meters, which is roughly equivalent to two seconds of free-falling.



Figure 34. The human-sized dummy that was used during the test throws.

For the needs of the experiment, we made 320 test throws of the dummy, to simulate a manoverboard event [see **Figure 35**(a)-(d)]. Additionally, we recorded several videos without dropping the dummy as well as numerous throws of various objects, such as plastic bags and bottles [see **Figure 35**(e)-(f)]. This way we can implement deep learning models that are not prone to false-positive alarms, triggered by non-human-related events.



(a) (b) (c) (d) (e) (f) **Figure 35.** Test throws during the data collection experiments. The free fall (a)-(d) of the human dummy from different shooting angles (positive event), and various other objects such as (e) plastic bags and (f) bottles (negative event).

The experiments took place in the surrounding area of Nikaia Olympic Weightlifting Hall, and lasted five days. Due to the fact that the test throws were carried out throughout the whole day, from 9:00 AM to 5:00 PM, the acquired videos vary in terms of illumination conditions (e.g., underexposure, overexposure). Additionally, we shot under various weather conditions (e.g., sunny, cloudy, rainy, windy, hot, cold), thus providing further variations in the background of the event.

Here, we are using a dataset consisted of RGB videos featuring the free falls of the dummy (see **Figure 35**(a)-(d)). For the dataset collection, which contains video sequences with a resolution of 1080×1920 pixels, we used a GoPro Hero 7 Silver (see **Figure 36**). The camera was set to shoot at a high frame rate, at 50 frames per second, to ensure sufficient acquisition of data that concerns the critical event. The dataset of this work is available online at: <u>https://github.com/ikatsamenis/Fall-Detection/</u> (accessed date 20 September 2022).



Figure 36. The RGB optical sensor, which was used during the data acquisition experiments to monitor the test throws of the human dummy, mounted on the building.

It is underlined that to avoid training bias and guarantee replicability of the results to other datasets, we placed the sensor in four different locations of the building, in order to obtain data that vary in terms of background, illumination, shooting angle, and distance [see **Figure 35**(a)-(d)]. In particular, as depicted in **Figure 37**, we placed the RGB camera (i) on the left of the fall at a close distance of 7m [see **Figure 35**(a)], (ii) on the right of the fall at a close distance of 5m [see **Figure 35**(b)], (iii) on the top left of the fall at an angle of roughly 45° [see **Figure 35**(c)], and (iv) to the left of the fall at a long distance of 13m [see **Figure 35**(d)]. It is emphasized that to further generalize the learning procedure, we augmented the training data by horizontally flipping the corresponding videos.



Figure 37. The four locations of the building where the optical sensor was placed, during the data acquisition experiments.

7.2.2. Model Training

The proposed method was implemented in the interactive environment called "Google Colaboratory", which allows the user to write Python codes through a browser. In this environment, important libraries are already installed, such as Tensorflow and Keras. This specific implementation used Python 3.7.12, Keras (1.08), and Tensorflow (2.1.0) machine learning libraries, in combination with various scientific and data management libraries. The model was trained using Tesla K80 GPU.

In order to train the model, a preprocessing stage was necessary. Preprocessing began with the separation of the RGB video data into the train and test set. No falling action data were used for the train set, while falling action data were used for the test set. Subsequently, frames were exported from the video data. These frames were resized and turned into gray scale, in order to train the autoencoder model.

Then, the training process was initiated by only using the data that had no falling action. These data constituted normal data. The test data were used for predictions after the training process. In order to study the most useful camera placement, two models were trained; the first model was designed for the horizontal view and the second one for the 45-degree angle view of the camera.

For comparison purposes, a supervised learning method was created, which consisted of a classifier code. In this method, the same data as in the unsupervised learning method were used, but the falling and no falling data were combined to the training process. More specifically, 60% of the entire dataset was used for the train set, 30% for the test set and 10% for the validation. In this method, the same preprocessing concept was followed and the focus was on the best camera placement, as in the unsupervised learning method.

The performance of the proposed method was tested in the dataset described in section 4.1. We started with a simple autoencoder over the appearance property, and tested its performance from multiple angles and compared it with a simple CNN classifier. For this purpose the model was trained on videos representing the normal condition, i.e. falls with zero numbers of falls in them. The testing of the performance took place using the falls and an equal number of frames depicting the normal condition.

The Area Under Curve (AUC) metric was employed in evaluating the performance of the proposed method. The AUC is computed with regard to ground-truth annotations at the frame-level and it is a common metric for many abnormal event detection methods. In this work, it was used to measure the ability of the learning algorithm to correctly distinguish falling from no falling events and summarize the Receiver Operating Characteristic (ROC) curve of the system. The ROC curve constitutes the probability curve that plots the raising of a true alert (true positive rate) and a false alarm (false positive rate) at various thresholds. The proposed algorithm achieved an AUC score of 100% for the horizontal view model and 59% for the 45-degree angle view model. The horizontal view model showed an excellent measure of separability. On the other hand, the 45-degree angle view model showed no class separation capacity. The AUC score proves that the horizontal view is the most suitable placement for the camera. The performance of the system using these metrics can be viewed in Figure 38 and Figure 39.



Figure 38. Autoencoder ROC Curve Sideways Camera



Figure 39. Autoencoder ROC Curve Top Camera

Autoencoder	Accuracy	Recall	Precision	F1
horizontal view	0,613475	0,613475	0,782	0,545585
45-degree angle view	0,5	0,5	0,25	0,333333

Table 9. Performance of the unsupervised autoencoder approach for the different positions of the camera

Table 10. Performance of a supervised classifier for the different positions of the camera

Classifier	Accuracy	Recall	Precision	F1
horizontal view	0,428571	0,375	0,25	0,3
45-degree angle view	0,5	0,5	0,25	0,33333

Metrics which consisted of accuracy, recall, precision and F1 score, were employed for the evaluation of the two methods. In order to compute these metrics, it was necessary to calculate True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values. These values can be displayed through the Confusion Matrix. Confusion Matrix is a special table layout that allows visualization of the method performance. Each row of the matrix represents the instances in an actual class and each column represents the instances in a predicted class.

Regarding the autoencoder metrics, the low percentage of the metrics lies in the fact that the labeling rate is low. The fact that the precision metric is high-scoring shows that there is a negligible quantity of FP values, which means that we had the minimum amount of false alarms. Concerning the placement of the camera, the horizontal view has proved to be the most suitable.

It is clear, considering the low percentage of the metrics, that a supervised learning method is inadequate for the purpose of this application scenario. The performance of the metrics is shown in Table 9 and Table 10. The comparison of the performance of the classifier and the autoencoder can be seen in **Figure 40**.



Figure 40. Comparative analysis unsupervised vs supervised approach for both capturing angles

From the analysis above, we see that an autoencoder model analyzing streams from the horizontal view angle, we provide the optimal results. These however still fail to achieve performance that can be considered sufficient for using it in real world scenarios. To this end, we mobilise an additional set of autoencoders over the additional image properties as seen in **Figure 41**. This increases the AUC score significantly, achieving and AUC of **97.3**. Based on the same annotation that was used for the comparative analysis of the autoencoder and the classifier in **Figure 40** we can assess the performance of the multiple autoencoder method. This can be seen in **Figure 42**.



Figure 42. Performance of multi-autoencoder approach

7.3. Conclusions

In this chapter, man overboard detection was formulated as an anomaly detection problem. We presented and evaluated an unsupervised learning algorithm for the automated recognition of such critical events, which is based on a spatiotemporal convolutional autoencoder. The employed technique models the normal conditions of the perimeter of the ship by learning the spatial and temporal features from the input video frames during the training stage and then identifies falls as abnormal behavior.

More specifically, the proposed framework uses multi-property (i.e., appearance, gradient, and saliency) analysis of RGB video streams in order to extract salient features and encodings of the normal scene utilizing a set of spatiotemporal convolutional autoencoders. Subsequently, the system can recognize a man overboard situation depending on whether the autoencoder is able or not to reconstruct a scene due to the potential existence of an abnormal event. Furthermore, to train and evaluate the performance of the proposed method, a dataset containing RGB video sequences with test throws of a human-sized dummy from the balcony of a high-rise building was demonstrated. The proposed multi-property spatiotemporal autoencoder achieved state-of-the-art results and, in particular, 97.30% accuracy and 96.01% F1-score on the test set of the presented dataset, surpassing other state-

of-the-art approaches, such as a single autoencoder, over the appearance property and a conventional CNN classifier. This entails a relative change in the error rate of 93.01% and 87.23% in terms of the accuracy and the F1-score, respectively. Therefore, through the proposed expansion of the autoencoder in such a way that it utilizes multiple image properties, the obtained error rate was roughly decreased to 1/10 of its original value.

8. Unsupervised Approach: Outlier detection in datasets including numerous simultaneous actionss

As stated before, even in the simple binary classification of actions (such as a normal abnormal paradigm presented in the previous chapter), there is difficulty in that the definition of an abnormal event is not always clear. What is an abnormal event is vague and tough to model. For this reason, the abnormal event detection problem is modeled as outlier detector. The main difficulty of applying an outlier scheme for abnormal event detection is that these methods are usually adopt a common global model for representing the whole normal space. However, usually normality consists of several sub-activities each one having quite different characteristics. Therefore, their modelling through a common model is not efficient.

For the technique presented in this chapter, we handle the abnormal event detection problem as an unsupervised learning paradigm. However, the limitations of the unsupervised approaches for abnormal event detection are the following: First, the number of clusters that a normal space is partitioned to, is a priori given an assumption which it is not valid in reallife application scenarios. It is clear that the sub-activities of the normal space are application dependent and therefore the number of clusters are highly related with the scenario. Second, the models assume no interrelations for events across different clusters (the sub-activities of the normal space), conditions that are also not valid for real-life cases. To address these difficulties, in this chapter, we introduce a framework for intra and inter property (feature) encoding to take into account property interrelations. In particular, we adopt convolutional autoencoders for compressing the video information at different property (feature) dimensions. Then, we introduce unsupervised tensor-based models for compressing the inter-property information resulting in a more compact normal space representation, increasing, consequently, the abnormal event detection performance. The overall proposed architecture for the abnormal event detection scheme is presented in **Figure 43**.

The technique presented here uses a two-fold scheme towards unsupervised abnormal event detection; the *Intra and Inter-Property Encoding*. In this way, we eliminate

the correlated information within and across image property features of video frames. Intra property encoding is implemented through auto-encoders as in the previous chapter, while a novel tensor-based unsupervised learning model is utilized as far as inter-property encoding is concerned. The current approaches, such as the work of [53], adopts a simple concatenation mechanism for fusing the intra-property compressed latent features. However, such an approach inherently implies that each property representation is independent from each other, an assumption which it is not valid. For example, the gradient property is highly correlated with the appearance as well as the saliency property. To address this difficulty, in this chapter, we introduce an alternative approach for fusing the intra-property compressed latent features together using a tensor-based unsupervised learning model. Tensor-based learning i) addresses the assumption that the partitions of the normal event space are a priori known and ii) reduces the dimensionality of space removing the inter-relationships across different properties. Tensor learning compacts the normal space partitioning, increasing the performance and generalization of the abnormal event detection. **Figure 43** presents the proposed methodology consisting of two main parts; *the intra and inter property encoding*.



Figure 43. Proposed twofold architectures for abnormal event detection

8.1. Intra-Inter property encoding

8.1.1. Property Representations

Let us first denote as $I \in \mathbb{R}^{N \times M}$ an image frame of $N \times M$ dimension. Let us also denote as $I_s \in \mathbb{R}^{N \times M \times k}$ a sequence of *k* consecutive image frames of *I*. A property, in this research, refers to a two-dimensional image operator applied on the stacked of image frames I_s generating an

image cube as its output. In other words, a property refers to a feature of an image sequence capable of transforming the raw image pixels into a more meaningful semantic information. We denote as $P_i \in \mathbb{R}^{N \times M \times k}$ the property image cube (e.g., a 3D tensor) which is generated by applying the *i*-th property operator on I_s . Let us also denote as *K* the number of image property operators used. Here, three image properties are considered (K = 3); the *appearance*, the *gradient* and the *saliency*. These properties are the same that were described in the previous chapter, i.e. Appearance (actual video frame), Motion (gradient of the frame), and Objectness (saliency map of the frame).

8.1.2. Intra Property Encoding using spatiotemporal autoencoders

The first part of the proposed methodology includes a set of convolutional autoencoders each associated for an image property. The purpose of these auto-encoders is to reduce the redundant information of a property extracting key property components in a hidden (latent) way. Here, three image properties are considered; *the appearance, the gradient and the saliency*.

The first two property features are in a similar line with previous works such as of [96], while saliency property is extracted to make our abnormal event detector more generic to different event types. The Appearance Property consists of the actual frame capturing. The Motion Property captures the movement of objects by taking as input the gradient of the frame. Finally, the Saliency Property reflects how likely a window of the frame covers an object of any category. This property creates a saliency map with the same size as the frame that covers all objects in an image in a category independent manner.

8.1.3. Inter Property Encoding using tensor-based unsupervised learning

The current approaches, such as the work of [53], adopt a simple mechanism for fusing the intra-property compressed latent features $C_k(L)$, by just concatenating the derived codebooks one after the one. However, such an approach inherently implies that each property representation is independent from each other, an assumption is not valid. For example, the gradient property is highly correlated with the appearance as well as saliency property. To address this difficulty, we introduce an alternative approach for fusing the intra-property compressed latent features as the outer product across all the compressed codebooks $C_k(L)$.

In particular, let us first vectorize each $C_k(L)$ and let us denote this vectorized signal as x_k , $k = 1, \dots, K$. We recall that three property operators are considered, and thus K = 3. Then, the fused property feature

$$\mathbf{X} = \mathbf{x}_1 \circ \mathbf{x}_2 \circ \cdots \circ \mathbf{x}_K \tag{4}$$

is derived as the outer product over all x_k . Therefore $X \in \mathbb{R}^{d_1 \times \cdots \times d_k}$, where d_k is the number of elements of the vectorized signal x_k .

While the outer product generates all possible correlations among the compressed property features (and therefore, it handles the issue of inter-relationships among them), it has the limitation of producing quite large tensors of high redundant information, confusing the direct application of an unsupervised clustering algorithm (e.g., *c*-means) for normal space partitioning. To overcome this difficulty, we introduce a novel tensor based unsupervised learning, with the main purpose of compressing tensor X.

The Inter-Property encoding part is also an autoencoding structure. The main difference is that we now involve nonlinear neuron operators, implementing as an inner product of the neuron weights and the input signal instead of convolutions. This is mainly due to the fact that the convolutional kernels are more suitable for processing image data. Instead, the neuron operators are more suitable for processing tensorbased data [96] as X.

Therefore, the inter-property encoding model consists of an inner-product tensor autoencoder where its input and output coincide with the tensor X. Below we define the tensor algebra operations utilized by the tensor autoencoder and then we describe rigorously the autoencoder's architecture.

Mode-*n* product. The mode-*n* product, $C = X \times_n B$ of a tensor $X \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ and a matris $B \in \mathbb{R}^{q \times d_n}$ yields a tensor $C \in \mathbb{R}^{d_1 \times \cdots \times d_{n-1} \times q \times d_n + 1 \times \cdots \times d_K}$.

Tucker decomposition. The Tucker decomposition provides a factorization of a tensor $X \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ into a core tensor $G \in \mathbb{R}^{q_1 \times \cdots \times q_k}$ and factor matrices $B^{(n)} = [b_1^{(n)}, b_2^{(n)}, \dots, b_1^{(q_n)}] \in \mathbb{R}^{d_n \times q_n}$, and can be expressed as follows:

$$X = G \times_1 B^{(1)} \times_2 B^{(2)} \dots \times_K B^{(K)} = \sum_{i=1}^{q_1} \sum_{i_K=1}^{q_K} g_{i_1 \dots i_K} \left(b_{i_1}^{(1)} \circ b_{i_2}^{(2)} \circ \dots \circ b_{i_K}^{(K)} \right) (5)$$

where $g_{i_1\cdots i_K}$ is the element of the core tensor G indexed by i_1, i_2, \cdots, i_K .

The tensor autoencoder is a fully connected feedforward neural network, however, its weights at each layer satisfy a Tucker decomposition; see (5). n particular, the weights W_l of the l-th hidden layer can be expressed as

$$W_{l} = I_{l} \times_{1} W_{l}^{(1)} \times_{2} W_{l}^{(2)} \dots \times_{K} W_{l}^{(K)}$$
(6)

where I is the core tensor all elements of which are equal to 1.

The information is propagated through the layers of the tensor autoencoder in a sequence of projections and nonlinear transformations. Due to (6), the tensor autoencoder at each layer projects tensor objects from a tensor space to another tensor space. Formally, a tensor sample $X \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ is projected to another tensor space by

$$Z_{1} = \mathcal{X} \times_{1} (W_{l}^{(1)})^{T} \times_{2} (W_{l}^{(2)})^{T} \dots \times_{K} (W_{l}^{(K)})^{T} (7)$$

where

$$W_l^{(k)} \in \mathbb{R}^{q_k^{(1)} \times d_k}$$
 and $Z_1 \in \mathbb{R}^{q_1^{(1)} \times q_2^{(1)} \times \dots \times q_k^{(1)}} \cdot q_j^{(l)}$

is the rank of the decomposition along mode k and the superscript denotes that this decomposition takes place on the *l*-th layer of the tensor autoencoder. Then Z₁ passes through the first hidden layer of the autoencoder, which applies the following nonlinear transformation

$$H_1 = \sigma(Z_1) \tag{8}$$

on it. The same pattern is used to propagate the information from the *l*-th hidden layer to the next one. Initially, the tensor object H_l is projected to another tensor space by $Z_{l+1} = H_l \times_1 (W_{l+1}^{(1)})^T \times_2 (W_{l+1}^{(2)})^T \dots \times_K (W_{l+1}^{(K)})^T$ and then the nonlinear transformation $H_{l+1} = \sigma(Z_{l+1})$ is used to produce the output of the (l + 1)-th hidden layer.

Let's assume that the autoencoder has L hidden layers. Then the dimension of the weights $W_l^{(k)}$ for the last hidden layer will be $(d_k \times q_k^{(L)})$ for $k = 1, \dots, K$. This way the input and the output of the tensor autoencoder are forced to have the same dimension. Let us denote

by X the output of the tensor autoencoder. Then, its weights are optimized via backpropagation by minimizing the following reconstruction error

$$E_t = //X - X//^{-2}, \qquad (9)$$

where $|| \cdot ||_2$ stands for the mean squared error.

The tensor autoencoder consists of two parts: the encoder and the decoder. The encoder, from layer to layer, reduces the dimension of the input, while the decoder increases the dimension so as the output of the autoencoder has the same dimension as the input. This way the autoencoder produces an information bottleneck which, when used in conjunction with the loss in (9), forces the encoder to learn input representations in a lower dimension that capture the most important aspects of input's information.



Figure 44. The tensor based learning algorithm adopted in the unsupervised tensor based network

8.1.4. Unsupervised Tensor –based Learning

Let us assume that we form a neural network-based auto-encoder, in which its inputs/outputs coincide with tensors X. Each neuron implements a non-linear relationship $g(\cdot)$, relied on the sigmoid function. We also assume that we have Q neurons at the hidden layer. The input X is weighted through parameters w_i and the inner product $\langle w_i, X \rangle$ is given as input to $g(\cdot)$. The response of the *i*-th hidden neuron is

$$u_i = g(\langle w_i, X \rangle) \quad (2)$$

weights w_i are tensors since the input X is a hyper-cube.

In Eq. (2), tensor u_i is a transformed version of X at the *i*-th hidden neuron. The decoder part receives as input the compressed signal u_i and transforms it to an output signal which should

be as close as possible to X. In the decoder, tensor u_i are first weighted by parameters v and then are inputted to neurons to generate an estimate \hat{X} of X.

$$\hat{X} = \langle y, g(\langle v, u_i \rangle) \rangle$$
 (3)

In Eq. (3), y denotes the parameters that weigh the outputs of the decoder to produce estimates of X. Since the network weights are huge due to the outer product, a tensor-based unsupervised learning is proposed for reducing significantly its parameters and consequently the number of data samples.

8.2. The Rank-1 Canonical Decomposition of Network Parameters

Let us assume that the weights w_i are rank-1 canonically decomposed into the weights $w_i^1, w_i^2, ..., w_i^D$, where w_i^D refers to the *D*-th rank-1 canonical decomposition of the weight w_i . Therefore, we have that

$$w_i = w_i^D \otimes \dots \otimes w_i^1 \quad (4)$$

In Eq. (4), the \otimes refers to the Kronecker product of the tensors $w_i^1, w_i^2, \dots, w_i^D$. Using tensor algebra, the inner product of $\langle w_i \cdot X \rangle$ can be written as

$$\langle w_i, X \rangle = \langle w_i^D \otimes ... \otimes w_i^1, X \rangle = \langle w_i^l, X_{\neq l} \rangle$$
 (5)

where $X_{\neq l}$ is a transformed version of the input signal X independent from the *l*-th rank-1 canonical decomposition w_i^l . More specifically, the $X_{\neq l}$ is given as

$$X_{\neq l} = X(w_i^D \odot \dots w_i^{l+1} \odot w_i^{l-1} \dots \odot w_i^1) \quad (6)$$

In Eq. (6) the \bigcirc denotes the Khatri-Rao product in tensor algebra. Using Eq. (5) and (6) one can re-write the encoding part of Eq. (2) as

$$u_i = g(\langle w_i, X \rangle) = g(\langle w_i^l, X_{\neq l} \rangle)$$
(7)

In a similar way, we can re-write the decoding part of the network using rank-1 canonical decomposition.

8.2.1. The Learning Algorithm

Using Eq. (7) we are able to train the network with a significant reduction in the number of its parameters. We initially fix all the weights $w_i^1, w_i^2, ..., w_i^D$ apart from the *l*-th. This way, the transformed version $X_{\neq l}$ is computer from Eq. (6). Then, using the backpropagation algorithm, we update only the weight w_i^l to minimize the error so that network output resembles as much as possible the respective inputs. Therefore, network parameters are solved in an iterative way with respect to one of the *D* canonical decomposed weight vectors, assuming the remaining fixed.



Figure 45. Our approach for abnormal event detection as outliers of normal space partitioning by the unsupervised tensor learning.

The output of the encoding part of the unsupervised tensor-based learning module is used to partition the normal activity space into sub-groups. This is depicted in **Figure 46**. Therefore, a way for detecting an abnormal event detection compared with a normal activity is to compare the event with respect to its distance to the normal activity space. In case that the reconstructed error with respect to the normal activity subgroups (representing by the tensors u_i) is high the event is considered not normal and therefore abnormal.

8.3. Experimental evaluation

The proposed method was tested using two popular benchmarking datasets, namely the Avenue [97] and Shanghai Tech [103]. The Avenue dataset includes 16 training videos and a total of 15,328 frames as well as 21 test videos or 15,324 test frames. For each frame ground truth locations of anomalies are provided. The Shanghai Tech dataset consists of 330 training and 107 testing videos. It contains of about 130 abnormal events.

The proposed method was implemented in Python. The autoencoders that implement the feature extraction (Appearance, Gradient and Saliency) were implemented in Tensorflow and Keras, while the tensor based autoencoder was implemented in PyTorch using the Tensorly library. The hyperparameter optimization of the learning algorithms was determined using the Hyperband optimization method of [104], which employs a principled early-stopping strategy to allocate resources, allowing it to evaluate orders-of-magnitude more configurations than black-box procedures like Bayesian optimization methods [105].

Method	Avenue Dataset	Shanghai Tech Dataset
Lu et al. [97]	80.9	-
Hasan et al. [100]	70.2	60.9
Del Giorno et al.	78.3	-
Smeureanu et al. [106]	84.6	-
Ionescu et al. [102]	80.6	-
Luo et al. [103]	81.7	68.0
Liu et al. [101]	85.1	72.8
Liu et al. [107]	84.4	-
Sultani et al. [108]	-	76.5
Ionescu et al. [98]	90.4	84.9
Our Method	86.9	79.8

Table 11. Abnormal Behavior detection based on frame level AUC on the Avenue and Shanghai tech datasets.

The Area Under Curve (AUC) metric was employed in assessing the performance of the proposed method and the compared ones. The AUC is computed with regard to ground-truth annotations at the frame-level and it is a common metric for many abnormal event detection methods. The performance comparison of our method with other implementations is presented in Table 13. For each of the compared methods, we choose the optimal parameter selection and thus the worst-case comparison scenario for our case. As we can see in the table above our method outperforms all nine works but one technique. Only [98] performs better. However, [98] employs an optimized k-means clustering on these datasets the

generalization of which to another data sequence is doubtful due to its limitations in initial condition selection, well separable clustering property (k-means fails in complex non-linear cluster separation like a spiral) and the distance metric adopted. Moreover, [98] uses an initial object detection step for preprocessing. This allows only for the detection of abnormalities relevant to specific objects, such as humans, that can be identified by the object detection method, while also introducing a computational overhead as a result of the frame preprocessing. Instead, our approach can be generalized to any type of object classes, such as falling debris, natural disaster detection et which can be seen as abnormal events. Table 12 and Figure 40 indicate the limitation of [98] in using k-means for normal event space partitioning. It is clear that the number of clusters selected is highly related with the application scenario used. In this figure, we have implemented the approach of [98] without the use of the initial object detection algorithm for different numbers of clusters. This is the reason of why the results are not the same as Table 14, which they have been optimized for a particular dataset. As is observed, the maximum accuracy is achieved for different numbers of clusters between different datasets.

The work of [98]	Our Approach
Dependent on k-means performance	Independent from any clustering
Dependent on specific objects/events, mainly humans	It works for any type of objects and events
Computational overhead	No additional overhead

 Table 12. Research difference summary of our work with [98].



Figure 47. Performance difference between different number of k in the Shangai and Avenue Dataset.

This drawback is also illustrated by the introduction of noise in the input video stream. The multi-property processing and the frame wide analysis of our method results in robustness towards noise introduced to the stream. Such noise can be the result of poor visibility conditions. Figure 42 presents this comparison with our method and [98] in this aspect. The figure illustrates the variance of AUC scores as the input signal's SNR drops. The noise introduced is simple Gaussian noise



Figure 48. Performance difference between different levels of noise in the video stream, Avenue Dataset
The response of our system to various abnormalities in a test video can be viewed in Figure 44. In the figure we have averaged the reconstruction errors in batches of 10 frames, for presentation purposes. The frames above are representative of the state captured in the bounding boxes in the graph. The annotation of abnormalities comes from the ground truth dataset.



Figure 49. Captured abnormalities and system response (Avenue Dataset). Axis x presents the frame batch while axis y represents the average reconstruction error. Above the detected abnormalities the annotated ground-truth data is presented

8.1. Conclusions

In this chapter, we introduce a novel method for abnormal event detection in video systems based on an intra/inter property feature information redundancy reduction. Intra property redundancy reduction is carried out using auto-encoders while the inter property one through tensor-based learning to take into account all potential interrelations of them. Experiments on benchmarked datasets show that our scheme outperforms all the compared works but one.

9. Conclusions

In this dissertation we presented the use of six different methods based on deep learning architectures for the analysis of visual data inside and outside the visible spectrum. Two application scenarios are considered, one where there is a priori knowledge of the captured actions, and one where the actions captured are unknown.

For the first application scenario we showcase that for the analysis inside the visible spectrum, the combination of the feature extraction capabilities of CNN architectures in combination with the temporal analysis abilities of LSTM networks achieves state of the art performance. However, due to particularities of the application scenario, adaptations on these architectures are required. Specifically, because of short-term dependencies in the classification of choreographic motion primitives, a memory window in the input layer, enhances the performance and allows for the output to change in the appropriate degree of granularity. Moreover, since the classification step is affected not only from previous but also the next states, the use of a bidirectional LSTM, increases the performance by inherently taking into account the non-causality of the input stream.

For the analysis outside the visible spectrum, the extraction of skeletal data from infrared depth sensors is a useful preprocessing step. This preprocessing transforms raw spatial data into semantically enriched structures that can be used for the classification, transforming the problem in a time-series analysis problem. Then, the use of a bidirectional LSTM network, enhanced with both autoregressive and moving average functionalities can successfully drive the classification step, achieving state of the art performance.

Additionally, since the selected application scenario has small datasets, and due to the known fact that deep learning techniques require extreme numbers of examples to achieve sufficient performance, we study the use of tensor-based learning networks for the classification of motion primitives in dance choreographies. We showcase that a tensor based network can achieve performances similar to the ones achieved by known state of the art classifiers, but using significantly less trainable parameters. This results in the ability of such tensor-based neural networks to be trained using smaller datasets, facilitating faster deployment and analysis.

For the second application scenario, where there are unknown actions inside the under analysis data, we present two main techniques. The first one is a supervised learning method where a deep NARMA filter, in the form of an adaptive CNN, achieves high classification performance. The proposed architecture also allows for the incorporation of additional data modalities in parallel to the visual ones, and the simultaneous analysis of these multi-modal data.

However, because of the lack of generalization of such supervised approaches, unsupervised ones are also studied. Initially, a convolutional spatiotemporal autoencoder is used to detect outlier actions. In this case, a fall detection problem is considered. The advantages of this technique is that it does not need the outlier action inside the training set. Instead, the autoencoder is only trained using samples from normal conditions. The appearance of an outlier action significantly affects the performance of the autoencoder, which is showcased in the reconstruction error of the decoder. We achieve extremely high performance in fall detection by employing this scheme in multiple parameters of visual data both inside and outside the visible spectrum.

Finally, we extend the autoencoder method, in order for it to be used in benchmarking datasets with large number of actions captured. The use of convolutional autoencoders in multiple visual properties allows for the proper modelling of all aspects of normality inside the dataset. Then a tensor-based autoencoder is used for effectively minimizing the dimensions of the normal state, which is then used for identifying outlier actions.

Future steps of the research presented in this dissertation include the use of architectures such as the newly published visual transformer layers to drive the classifications. Moreover, and specifically for the technique presented in chapter 8, beyond the monitoring of the autoencoder reconstruction error, can be used as classifications. An example of that would be the breaking down of the normal state into multiple normality substates, and then use appropriate classifiers, or even conformal learning schemes, to allow for

better performance. Finally, the use of all this techniques can be greatly enriched by using techniques to not only classify the data, but also explain the reasons behind the classification outcome. Such explainable AI techniques can facilitate the use of such tools not only from researchers that are well versed in machine learning algorithms, but also by multidisciplinary teams that can also bring domain specific knowledge into the analysis.

References

- Ben-Arie, J., Wang, Z., Pandit, P., & Rajaram, S. (2002). Human activity recognition using multidimensional indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8), 1091-1104.
- [2] Chéron, G., Laptev, I., & Schmid, C. (2015). P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 3218-3226).
- [3] Hadfield, S., & Bowden, R. (2013). Hollywood 3D: Recognizing actions in 3D natural scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3398-3405).
- [4] Fan, Y., Lu, X., Li, D., & Liu, Y. (2016, October). Video-based emotion recognition using CNN-RNN and C3D hybrid networks. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction* (pp. 445-450). ACM.
- [5] Piccardi, M. (2004, October). Background subtraction techniques: a review. In 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583) (Vol. 4, pp. 3099-3104). IEEE.
- [6] Milbich, T., Bautista, M., Sutter, E., & Ommer, B. (2017). Unsupervised video understanding by reconciliation of posture similarities. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4394-4404).
- [7] Rallis, I., Doulamis, N., Doulamis, A., Voulodimos, A., & Vescoukis, V. (2018). Spatiotemporal summarization of dance choreographies. Computers & Graphics, 73, 88-101.
- [8] Wang, H., Kläser, A., Schmid, C., & Cheng-Lin, L. (2011, June). Action recognition by dense trajectories. In CVPR 2011-IEEE Conference on Computer Vision & Pattern Recognition (pp. 3169-3176). IEEE.
- [9] Kolekar, M. H., & Dash, D. P. (2016, November). Hidden markov model based human activity recognition using shape and optical flow based features. In 2016 IEEE Region 10 Conference (TENCON) (pp. 393-397). IEEE.

- [10] Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: a brief review. Computational intelligence and neuroscience, 2018.
- [11] Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P., & Zhang, J. (2014, November). Convolutional neural networks for human activity recognition using mobile sensors. In 6th International Conference on Mobile Computing, Applications and Services (pp. 197-205). IEEE.
- [12] Khaire, P., Kumar, P., & Imran, J. (2018). Combining CNN streams of RGB-D and skeletal data for human activity recognition. *Pattern Recognition Letters*, *115*, 107-116.
- [13]Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems* (pp. 568-576).
- [14] Wang, L., Ge, L., Li, R., & Fang, Y. (2017). Three-stream CNNs for action recognition. *Pattern Recognition Letters*, 92, 33-40.
- [15]Kamel, A., Sheng, B., Yang, P., Li, P., Shen, R., & Feng, D. D. (2018). Deep convolutional neural networks for human action recognition using depth maps and postures. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, (99), 1-14.
- [16] Makantasis, K., Doulamis, A., Doulamis, N., & Psychas, K. (2016, September). Deep learning based human behavior recognition in industrial workflows. In 2016 IEEE International Conference on Image Processing (ICIP) (pp. 1609-1613). IEEE.
- [17]Gan, C., Wang, N., Yang, Y., Yeung, D. Y., & Hauptmann, A. G. (2015). Devnet: A deep event network for multimedia event detection and evidence recounting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2568-2577).
- [18] Makantasis, K., Doulamis, A., & Doulamis, N. (2013, July). Vision-based maritime surveillance system using fused visual attention maps and online adaptable tracker. In 2013 14th international workshop on image analysis for multimedia interactive services (WIAMIS) (pp. 1-4). IEEE.
- [19] Babaee, M., Dinh, D. T., & Rigoll, G. (2018). A deep convolutional neural network for video sequence background subtraction. *Pattern Recognition*, 76, 635-649.

- [20] Varadarajan, S., Miller, P., & Zhou, H. (2015). Region-based mixture of gaussians modelling for foreground detection in dynamic scenes. *Pattern Recognition*, 48(11), 3488-3503.
- [21] Liang, X., Liao, S., Wang, X., Liu, W., Chen, Y., & Li, S. Z. (2018, July). Deep background subtraction with guided learning. In 2018 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1-6). IEEE.
- [22] R. Taormina, S. Galelli, N.O. Tippenhauer, E. Salomons, and A. Ostfeld.
 "Characterizing cyber-physical attacks on water distribution systems." *Journal of Water Resources Planning and Management Division*, ASCE, 04017009-1 04017009-12, 2017.
- [23] M. Housh and Z. Ohar. "Model-based approach for cyber-physical attack detection in water distribution systems." *Water Research*, Vol. 139, pp. 132-143, 2018.
- [24] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE CVPR, Las Vegas, NV, 2016, pp. 779-788.
- [25] Hai Zhu, Fu Xiao, Lijuan Sun, Ruchuan Wang, and Panlong Yang, "R-TTWD: Robust Device-Free Through-The-Wall Detection of Moving Human with WiFi", *IEEE Journal on selected areas in communications*, vol. 35, no. 5, May 2017.
- [26] K. Makantasis, A. Nikitakis, A. Doulamis, N. Doulamis and Y. Papaefstathiou, "Data-Driven Background Subtraction Algorithm for in-Camera Acceleration in Thermal Imagery," in *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.
- [27] S. Herrero and J. Bescs. "Background subtraction techniques: Systematic evaluation and comparative analysis," 11th International Conference on Advanced Concepts for Intelligent Vision Systems, ser. ACIVS '09. Springer-Verlag, 2009.
- [28] D. S. Yeo, "Superpixel-based tracking-by-segmentation using markov chains.," *IEEE Conference in Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [29] D. Kosmopoulos, A. Voulodimos, A. Doulamis, "A system for multicamera task recognition and summarization for structured environments," *IEEE Transactions on Industrial Informatics*, 9 (1), 161-171, 2013.

- [30] H.M. Mousavi, "Analyzing tracklets for the detection of abnormal crowd behavior," *IEEE Winter Conference on In Applications of Computer Vision (WACV)*, 2015.
- [31]S. A. Ahmed, D. P. Dogra, S. Kar and P. P. Roy, "Trajectory-based surveillance analysis: A survey," in *IEEE Trans. on Circuits and Systems for Video Technology*. [Available online: July 2018]
- [32] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 7068349, 13 pages, 2018 https://doi.org/10.1155/2018/7068349.
- [33] M. Youssef, M. Mah, and A. Agrawala, "Challenges: Device-free passive localization for wireless environments," in *Proc. ACM MobiCom*, 2007, pp. 222–229.
- [34] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. M. Ni, "FILA: Fine-grained indoor localization," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2210–2218.
- [35] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, p. 53, 2011.
- [36] S. Amin, X. Litrico, S. Sastry, A.M. Bayen, "Cyber security of water SCADA Systems-Part II: attack detection using enhanced hydrodynamic models." *IEEE Trans. Contr. Syst. Technol.* 21 (5), 1679-1693.
- [37] R. Taormina, S. Galelli, N.O. Tippenhauer, E. Salomons, "The battle of the attack detection algorithms," *Journal of Water Resources Planning and Management Division*, ASCE, 2018.
- [38] N. Nicolaou, D. Eliades, C. Panayiotou, and M. Polycarpou, "Reducing vulnerability to cyber-physical attacks in water distribution networks." *Intl Workshop on CySWater*, Porto, 2018.
- [39] D. Jiang, D. Zhuang, Y. Huang and J. Fu, "Survey of multispectral image fusion techniques in remote sensing applications", *Image Fusion and its applications*, Y. Zheng, INTECH Open Access Publisher, Vol. 1, pp. 1-22, 2011.
- [40] A. R. Pal and A. Singha, "A comparative analysis of visual and thermal face image fusion based on different wavelet family," *2017 International Conference on Innovations*

in Electronics, Signal Processing and Communication (IESC), Shillong, 2017, pp. 213-218.

- [41] Walter, C.S.S., Silva, Y.M.L. and de Lucena Jr, V.F., 2017. A Location Technique Based on Hybrid Data Fusion used to Increase the Indoor Location Accuracy. *Procedia Computer Science*, 113, pp.368-375.
- [42] F. Garcia, D. Martin, A. De La Escalera, and J.M. Armingol, "Sensor fusion methodology for vehicle detection," *IEEE Intelligent Transportation Systems Magazine*, 9(1), pp.123-133, 2017.

[43] S. Lee, H. G. Kim and Y. M. Ro, "BMAN: Bidirectional Multi-Scale Aggregation Networks for Abnormal Event Detection," *IEEE Trans. on Image Proc.*, vol. 29, pp. 2395-2408, 2020.

[44] A.S. Voulodimos, N.D. Doulamis, D.I. Kosmopoulos, and T.A. Varvarigou, "Improving multi-camera activity recognition by employing neural network based readjustment," *Applied Artificial Intelligence*, 26(1-2), 97-118, 2012.

[45] N. Bakalos, et al. "Protecting water infrastructure from cyber and physical threats: Using multimodal data fusion and adaptive deep learning to monitor critical systems." *IEEE Signal Processing Magazine*, 36.2, pp. 36-48, 2019.

[46] S. Wan, X. Xu, T. Wang and Z. Gu, "An Intelligent Video Analysis Method for Abnormal Event Detection in Intelligent Transportation Systems," IEEE Trans. on Intell. Transportation Systems, (to be published)

[47] R. Leyva, V. Sanchez and C. Li, "Fast Detection of Abnormal Events in Videos with Binary Features," *IEEE ICASSP*, Calgary, AB, pp. 1318-1322, 2018.

[48] S. Yan, J. S. Smith, W. Lu and B. Zhang, "Abnormal Event Detection from Videos Using a Two-Stream Recurrent Variational Autoencoder," *IEEE Trans. on Cognitive and Developmental Systems*, vol. 12, no. 1, pp. 30-42, March 2020.

[49] X. Sun, S. Zhu, S. Wu and X. Jing, "Weak Supervised Learning Based Abnormal Behavior Detection," *24th International Conf. on Pattern Recognition (ICPR)*, Beijing, 2018, pp. 1580-1585.

[50] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Video anomaly detection and localization using hierarchical feature repre- sentation and Gaussian process regression," *IEEE CVPR*, pp. 2909–2917, 2015.

[51] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):18–32, 2014.

[52] C. Lu, J. Shi, and J. Jia, "Abnormal Event Detection at 150 FPS in MATLAB," *IEEE ICCV*, pages 2720–2727, 2013.

[53] R. T. Ionescu, F. S. Khan, M.I. Georgescu, and L. Shao, "Object-centric autoencoders and dummy anomalies for abnormal event detection in video," *IEEE CVPR*, pp. 7842-7851, 2019.

[54] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep- cascade: Cascading 3D Deep Neural Networks for Fast Anomaly Detection and Localization in Crowded Scenes," *IEEE Transactions on Image Processing*, 26(4):1992–2004, 2017.

[55] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," *IEEE CVPR*, pages 733–742, 2016.

[56] H. Ren, W. Liu, S. I. Olsen, S. Escalera, and T. B. Moes- lund, "Unsupervised Behavior-Specific Dictionary Learning for Abnormal Event Detection," *Proc. of BMVC*, pp. 28.1–28.13, 2015.

[57] Xu Dan, Ricci Elisa, Yan Yan, Song Jingkuan and Sebe Nicu, "Learning Deep Representations of Appearance and Motion for Anomalous Event Detection", BMVC, 2015.

[58] L. Wang, F. Zhou, Z. Li, W. Zuo and H. Tan, "Abnormal Event Detection in Videos Using Hybrid Spatio-Temporal Autoencoder," 25th IEEE International Conference on Image Processing (ICIP), Athens, 2018, pp. 2276-2280, 2018.

[59] M. Ravanbakhsh, M. Nabi, E. Sangineto, L Marcenaro, C. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," *IEEE ICIP*, pp. 1577-1581, sept. 2017.

[60] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection—A new baseline," *IEEE CVPR*, pp. 6536–6545, July 2018.

[61] S. Lee, H. G. Kim, and Y. M. Ro, "STAN: Spatio-temporal adversarial networks for abnormal event detection," *IEEE ICASSP*, pp. 1323–1327, Apr. 2018.

[62] C. Sun, Y. Jia, H. Song and Y. Wu, "Adversarial 3D Convolutional Auto-Encoder for Abnormal Event Detection in Videos," IEEE Transactions on Multimedia, (to be published).

[63] A. Del Giorno, J. Bagnell, and M. Hebert, "A Discrimina- tive Framework for Anomaly Detection in Large Videos," Proc. of ECCV, pp. 334–349, 2016.

[64] X. Mo, V. Monga, R. Bala, and Z. Fan, "Adaptive sparse representations for video anomaly detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 4, pp. 631–645, Apr. 2014.

[65] F. Jiang, Y. Wu, and A. K. Katsaggelos, "A dynamic hierarchical clustering method for trajectory-based unusual video event detection," *IEEE Trans. Image Process.*, vol. 18, no. 4, pp. 907–913, Apr. 2009.

[66] I. Rallis, N. Bakalos, N. Doulamis, A. Voulodimos, A. Doulamis and E. Protopapadakis, "Learning Choreographic Primitives Through A Bayesian Optimized Bi-Directional LSTM Model," 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 1940-1944, doi: 10.1109/ICIP.2019.8803118.

[67] N. Bakalos, I. Rallis, N. Doulamis, A. Doulamis, A. Voulodimos and E. Protopapadakis, "Adaptive Convolutionally Enchanced Bi-Directional Lstm Networks For Choreographic Modeling," 2020 IEEE International Conference on Image Processing (ICIP), 2020, pp. 1826-1830, doi: 10.1109/ICIP40778.2020.9191307.

[68] K. Makantasis, A. Voulodimos, A. Doulamis, N. Bakalos and N. Doulamis, "Space-Time Domain Tensor Neural Networks: An Application on Human Pose Classification,"
2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 4688-4695, doi: 10.1109/ICPR48806.2021.9412482.

[69] N. Bakalos et al., "Protecting Water Infrastructure From Cyber and Physical Threats: Using Multimodal Data Fusion and Adaptive Deep Learning to Monitor Critical Systems," in IEEE Signal Processing Magazine, vol. 36, no. 2, pp. 36-48, March 2019, doi: 10.1109/MSP.2018.2885359.

[70] Katsamenis, I.; Bakalos, N.; Karolou, E.E.; Doulamis, A.; Doulamis, N. Fall
 Detection Using Multi-Property Spatiotemporal Autoencoders in Maritime Environments.
 Technologies 2022, 10, 47. <u>https://doi.org/10.3390/technologies10020047</u>

[71] D. Baraff, "Rigid body simulation," in Proc. of the SIGGRAPH Course Notes, 1992, vol. 19, pp. 1–68.

- [72] A.D. Doulamis, N.D. Doulamis, and S. D. Kollias, "An adaptable neural-network model for recursive nonlinear traffic prediction and modeling of MPEG video sources," *IEEE Transactions on Neural Networks*, Vol. 14, No. 1, pp. 150-166, 2003.
- [73] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Pro- cessing, vol. 45, no. 11, pp. 2673–2681, 1997.
- [74] J. Hochreiter, S.and Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
- [75] N. Doulamis, A. Doulamis, C. Ioannidis, M. Klein, and M. Ioannides, "Modelling of static and moving objects: Digitizing tangible and intangible cultural heritage," In Mixed Reality and Gamification for Cultural Heritage, pp. 567–589, 2017.
- [76] K. Makantasis, A. Nikitakis, A. D. Doulamis, N. D. Doulamis and I. Papaefstathiou, "Data-Driven Background Subtraction Algorithm for In-Camera Acceleration in Thermal Imagery," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 9, pp. 2090-2104, Sept. 2018.
- [77] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with Deep Bidirectional LSTM," IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 273–278, 2013.
- [78] N. Bakalos, I. Rallis, N. Doulamis, A. Doulamis, E. Protopapadakis and A. Voulodimos, "Choreographic Pose Identification using Convolutional Neural Networks," 11th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games), pp. 1-7, Vienna, Austria, 2019.
- [79] Baihua Li and H. Holstein, "Recognition of human periodic motion-a frequency domain approach," 16th International Conference on Pattern Recognition, ICPR, pp. 311-314, Quebec, Canada, 2002.

- [80] N. Bakalos, I. Rallis, N. Doulamis, A. Doulamis, E. Protopapadakis and A. Voulodimos, "Choreographic Pose Identification using Convolutional Neural Networks," 11th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games), pp. 1-7, Vienna, Austria, 2019.
- [81] X. Li, D. Xu, H. Zhou, and L. Li, "Tucker tensor regression and neuroimaging analysis," Statistics in Biosciences, vol. 10, no. 3, pp. 520–545, 2018.
- [82] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," SIAM review, vol. 51, no. 3, pp. 455–500, 2009.
- [83] J. Kossaifi, Z. C. Lipton, A. Khanna, T. Furlanello, and A. Anandkumar, "Tensor regression networks," arXiv preprint arXiv:1707.08308, 2017.
- [84] G. Hu, Y. Hua, Y. Yuan, Z. Zhang, Z. Lu, S. S. Mukherjee, T. M. Hospedales, N. M. Robertson, and Y. Yang, "Attribute-enhanced face recognition with neural tensor fusion networks," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3744–3753.
- [85] A. Cichocki, A.-H. Phan, Q. Zhao, N. Lee, I. Oseledets, M. Sugiyama, D. P. Mandic et al., "Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives," Foundations and Trends in Machine Learning, vol. 9, no. 6, pp. 431–673, 2017.
- [86] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
- [87] K. Makantasis, A. Nikitakis, A. Doulamis, N. Doulamis and Y. Papaefstathiou, "Data-Driven Background Subtraction Algorithm for in-Camera Acceleration in Thermal Imagery," in IEEE Transactions on Circuits and Systems for Video Technology, 2017.
- [88] Hai Zhu, Fu Xiao, Lijuan Sun, Ruchuan Wang, and Panlong Yang, "R-TTWD: Robust Device-Free Through-The-Wall Detection of Moving Human with WiFi", IEEE Journal on selected areas in communications, vol. 35, no. 5, May 2017.

- [89] S. Palipana, P. Agrawal, D. Pesch, "Channel state information based human presence detection using non-linear techniques," *Proc. 3rd ACM BuildSys 2016*, pp. 177-186, 2016.
- [90] S. Palipana, P. Agrawal, D. Pesch, "Channel state information based human presence detection using non-linear techniques," *Proc. 3rd ACM BuildSys 2016*, pp. 177-186, 2016.
- [91] L. Davies and U. Gather, "The identification of multiple outliers," J. Amer. Statist. Assoc., vol. 88, no. 423, pp. 782–792, 1993.
- [92] V.M. Igure, S.A. Laughter, and R.D. Williams, "Security issues in SCADA networks," Computers & Security, 25(7), 498-506, 2006.
- [93] G. Cohen, G. Sapiro, R. Giryes, "DNN or k-NN: That is the Generalize vs. Memorize Question," arXiv:1805.06822.
- [94] Örtlund, E.; Larsson, M. Man Overboard Detecting Systems Based on Wireless Technology. Bachelor Thesis, Chalmers University of Technology, Gothenburg, Sweden, 2018.
- [95] Sevïn, A.; Bayilmi s, C.; Ertürk, I.; Ekïz, H.; Karaca, A. Design and Implementation of a Man-Overboard Emergency Discovery System Based on Wireless Sensor Networks. Turk. J. Electr. Eng. Comput. Sci. 2016, 24, 762–773.

[96] K. Makantasis, A. D. Doulamis, N. D. Doulamis and A. Nikitakis, "Tensor-Based Classification Models for Hyperspectral Data Analysis," IEEE Transactions on Geoscience and Remote Sensing, vol. 56, no. 12, pp. 6884-6898, Dec. 2018.

[97] C. Lu, J. Shi, and J. Jia, "Abnormal Event Detection at 150 FPS in MATLAB," IEEE ICCV, pages 2720–2727, 2013.

[98] R. T. Ionescu, F. S. Khan, M.I. Georgescu, and L. Shao, "Object-centric autoencoders and dummy anomalies for abnormal event detection in video," IEEE CVPR, pp. 7842-7851, 2019.

[99] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep- cascade: Cascading 3D Deep Neural Networks for Fast Anomaly Detection and Localization in Crowded Scenes," IEEE Transactions on Image Processing, 26(4):1992–2004, 2017.

[100] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," IEEE CVPR, pages 733–742, 2016.

[101] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection—A new baseline," IEEE CVPR, pp. 6536–6545, July 2018.

[102] R.T. Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, "Un-masking the abnormal events in video," IEEE ICCV, pp. 2895–2903, 2017.

[103] W. Luo, W. Liu, and S. Gao. "A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework," In Proceedings of ICCV, pages 341–349, 2017.

[104] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," The Journal of Machine Learning Research, 18(1), pp.6765-6816, 2016.

[105] Kaselimi, Maria, et al. "Bayesian-optimized bidirectional LSTM regression model for non-intrusive load monitoring." ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.

[106] S. Smeureanu, R. T. Ionescu, M. Popescu, and B. Alexe, "Deep Appearance Features for Abnormal Behavior Detection in Video," In Proceedings of ICIAP, Volume 10485, pages 779–789, 2017.

[107] Y. Liu, C.-L. Li, and B. Poczos, "Classifier Two-Sample Test for Video Anomaly Detections," In Proceedings of BMVC, 2018.

[108] W. Sultani, C. Chen, and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," In Proceedings of CVPR, pages 6479–6488, 2018.