Compression de données et apprentissage en profondeur pour les applications de santé IoT basées sur des signaux physiologiques

Joseph Azar

UBFC
UNIVERSITÉ
BOURGOGNE FRANCHE-COMTÉ

UFC
UNIVERSITÉ
DE FRANCHE-COMTÉ

THÈSE DE DOCTORAT

DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

PRÉPARÉE À L'UNIVERSITÉ DE FRANCHE-COMTÉ

École doctorale n°37

Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

par

JOSEPH AZAR

Data Compression and Deep Learning for IoT Healthcare Applications based on Physiological Signals

Compression de données et apprentissage en profondeur pour les applications de santé IoT basées sur des signaux physiologiques

Thèse présentée et soutenue à Belfort, le 09 Octobre 2020

Composition du Jury :

PROF ELIZONDO DAVID	Université De Montfort	Rapporteur
PROF FORESTIER GERMAIN	Université Haute-Alsace	Rapporteur
PROF EL FALLAH-SEGHROUCHNI AMAL	Université de Sorbonne	Examinatrice
PROF DEMERJIAN JACQUES	Université Libanaise	Examinateur
PROF GUYEUX CHRISTOPHE	Université Bourgogne Franche-Comté	Examinateur
PROF COUTURIER RAPHAËL	Université Bourgogne Franche-Comté	Directeur de thèse
PROF MAKHOUL ABDALLAH	Université Bourgogne Franche-Comté	Codirecteur de thèse

N° X X X

ABSTRACT

Data Compression and Deep Learning for IoT Healthcare Applications based on Physiological Signals

Joseph Azar
University of Bourgogne Franche Comté, 2020

Supervisors: Raphaël Couturier and Abdallah Makhoul

In recent years, Internet of Things (IoT) technology has gained tremendous attention for its ability to relieve the burden on healthcare caused by an aging population and the increase in chronic disease. IoT technology facilitates the tracking of patients with different conditions and the processing of vast volumes of data, of which a substantial part of this data are physiological signals. Physiological signals are an invaluable source of data which helps to diagnose, rehabilitate, and treat diseases. The signals come from a Wireless Body Sensor Network or wearable devices placed on a patient's body. There are many difficulties in IoT-healthcare systems, such as data collection and processing especially that (1) wireless sensor nodes have limited energy, processing and memory resources, (2) the quantity of data collected periodically is enormous, (3) the quality of the data collected is not always satisfactory, and that such data are highly likely to include noisy or unreliable areas, and (4) the manual feature extraction process from the physiological signals requires significant human intervention and medical expertise.

Firstly, an energy-efficient data compression technique is being proposed in this dissertation. The proposed scheme is based on an error-bound lossy compressor originally designed for high-performance computing applications and has been adapted for IoT devices that are resource constrained. The proposed solution is an easy-to-implement algorithm, which could reduce energy consumption by as much as 2.5 times. It also reduces the processing/transmission time by compressing large batches of data before their transfer from the IoT to the edge. Moreover, an empirical analysis was carried out to study the effect of lossy time series compression on the classification task. In addition to different variations of compression methods, various deep neural networks for time series clas-

sification were considered to identify the appropriate trade-off between the compression ratio and classification performance.

Second, the problem of data distortion due to lossy compression was addressed. The reconstructed data may not always be satisfactory, given that the physiological signals collected are multivariate time series that are highly compressed with a lossy compressor prior to transmission. To solve this limitation, a convolutional autoencoder-based deep learning model was introduced. The proposed model was able to enhance the compressed data after reconstruction, thus fixing the shape of the physiological signal and allowing more accurate feature extraction.

Then, the photoplethysmogram signals were given particular attention. Photoplethysmography (PPG) is used to measure the skin blood flow using infrared light. Photoplethysmography is a promising technique because of the capability of the new wrist-worn devices to provide the signal. The existence of motion artifacts and meaningless areas in the signal is a major challenge faced when working with this signal. A deep learning model for automatic motion artifacts detection based on a CNN-LSTM autoencoder architecture has been proposed to detect and discard irrelevant zones in photoplethysmogram signals to avoid analyzing and processing meaningless data.

Finally, a deep learning model has been proposed for time series classification. Given that hand-designing features from physiological signals is a challenging task, the representation learning approach is a potential solution for automatically learning features from raw physiological data. A classification model based on the DenseNet architecture was proposed for the classification of multivariate time series. The results show that the proposed approach is capable of achieving and in several cases bypassing the performance of the state-of-the-art models on a benchmark dataset.

KEYWORDS: Energy Consumption, Data compression, Digital signal processing, Internet of Things, Health informatics, Stress detection, Physiological signals, Time series classification, Deep learning, Data enhancement.

RÉSUMÉ

Compression de données et apprentissage en profondeur pour les applications de santé IoT basées sur des signaux physiologiques

Joseph Azar
Université de Bourgogne Franche Comté, 2020

Encadrants: Raphaël Couturier et Abdallah Makhoul

Ces dernières années, la technologie de l'Internet des objets (IoT) a suscité un vif intérêt en raison de sa capacité à alléger certaines tâches de soins de santé causé par le vieillissement de la population et l'augmentation des maladies chroniques. La technologie IoT facilite le suivi des patients souffrant de différentes pathologies et le traitement de vastes volumes de données, dont une partie substantielle de ces données sont des signaux physiologiques. Les signaux physiologiques sont une source inestimable de données qui aident à diagnostiquer, réhabiliter et traiter les maladies. Les signaux proviennent d'un réseau de capteurs corporels sans fil ou d'appareils portables placés sur le corps d'un patient. Il existe de nombreuses difficultés dans les systèmes de soins de santé IoT, tels que la collecte et le traitement des données. En particulier les problèmes suivants peuvent survenir : (1) les nœuds de capteurs sans fil ont des ressources énergétiques, de traitement et de mémoire limitées, (2) la quantité de données collectées périodiquement est énorme, (3) la qualité des données collectées n'est pas toujours satisfaisante et ces données sont susceptibles d'être bruitées ou peu fiables, et (4) le processus d'extraction manuelle des caractéristiques à partir des signaux physiologiques nécessite une intervention humaine et une expertise médicale importantes.

Tout d'abord, une technique de compression de données éco-énergétique est proposée dans cette thèse. Le schéma proposé est basé sur un algorithme de compression avec perte. Cet algorithme a été conçu à l'origine pour des applications de calcul haute performance et a été adapté pour les dispositifs IoT qui sont limités en ressources. La solution proposée est un algorithme facile à mettre en œuvre, qui peut réduire la consommation d'énergie jusqu'à 2,5 fois. Il réduit également le temps de traitement / transmission

en compressant les données avant leur transfert de l'IoT vers une machine Edge. De plus, une analyse empirique a été réalisée pour étudier l'effet de la compression des séries temporelles avec perte sur la tâche de classification. De plus, nous avons conçus différentes variantes des méthodes de compression et plusieurs réseaux de neurones profonds pour la classification des séries temporelles afin d'obtenir un compromis entre le taux de compression et les performances de classification.

Deuxièmement, le problème de la distorsion des données due à la compression avec perte a été étudié et résolu. Pour résoudre cette limitation, un modèle d'apprentissage profond basé sur un auto-encodeur convolutionnel a été conçu. Le modèle proposé a pu améliorer les données compressées après la reconstruction, fixant ainsi la forme du signal physiologique et permettant une extraction plus précise des caractéristiques.

Ensuite, les signaux du photopléthysmogramme ont fait l'objet d'une attention particulière. La photopléthysmographie (PPG) est utilisée pour mesurer le flux sanguin cutané à l'aide de la lumière infrarouge. L'existence d'artefacts de mouvement et de zones dénuées de sens dans le signal est un défi majeur rencontré lorsque vous travaillez avec ce signal. Un modèle d'apprentissage profond pour la détection automatique d'artefacts de mouvement basé sur une architecture d'autoencodeur CNN-LSTM a été proposé pour détecter et éliminer les zones non pertinentes dans les signaux de photopléthysmogramme afin d'éviter d'analyser et de traiter des données dénuées de sens.

Enfin, un modèle d'apprentissage profond a été proposé pour la classification des séries temporelles. Un modèle de classification basé sur l'architecture DenseNet a été proposé pour la classification des séries temporelles multivariées. Les résultats montrent que l'approche proposée est capable d'atteindre et, dans plusieurs cas, de dépasser les performances des meilleurs modèles sur un ensemble de données de référence.

Mots clés: Consommation d'énergie, Compression de données, Traitement numérique du signal, Internet des objets, Applications de santé, Détection du stress, Signaux physiologiques, Classification des séries temporelles, Deep learning, Amélioration des séries temporelles

ACKNOWLEDGEMENTS

I would like to thank my PhD advisors, Professors Raphaël Couturier and Abdallah Makhoul, for supporting me during these past three years. I am very grateful to Raphael for his scientific advice and knowledge and insightful discussions and suggestions. He is my primary resource for getting my scientific questions answered. I am also thankful to Abdallah for all the fruitful discussions we had and for being supportive. He has given me the freedom to pursue various projects without objection.

Besides my supervisors, I would like to give a special thanks to Professor Jacques Demerjian, my master's thesis supervisor, for his helpful career advice, support, and suggestions in general.

A special thanks to Dr. Hassan Noura, whom I met at the IWCMC conference 2018, has been a friend and a supporter since then. He has helped provide advice many times and collaborate with me on numerous projects.

My sincere appreciation and thanks to the members and professors of the team AND (Algorithmique Numerique Distribuée) for the positive work environment and for the pleasant moments that we have shared together. I want to express my gratitude to all the Ph.D. students at the Lab, with whom I spent beautiful moments during the past years.

I would also like to express my strong thanks to my dearest friends who supported me in the past three years: Nancy Awad, Anthony Nassar, Gaby Boutayeh, Lama Sleem, Charbel Alam, Christian Saleem, and Carole Habib.

On a personal level, I want to thank all the members of my big family. I especially thank my mom, dad, and sisters. I want to thank my close friend Tony and my beloved fiancée Cynthia for believing in me and providing continuous and immeasurable support.

CONTENTS

List of abbreviations

AAL . . . . . Ambient Assisted Living

AE . . . . . AutoEncoder

AI . . . . . Artificial Intelligence

ANN . . . . . Artificial Neural Network

AUC . . . . . Area Under the Curve

BLE . . . . . Bluetooth Low Energy

BN . . . . . Batch Normalization

CCA . . . . . Canonical Correlation Analysis

CNN . . . . . Convolutional Neural Network

COTE . . . . . Collective of Transformation Ensembles

CR . . . . . Compression Ratio

CS . . . . . Compressive Sensing

DAE . . . . . Denoising AutoEncoder

DBS . . . . . Deep Brain Stimulation

DenseNet . . . Densely Connected Convolutional Network

DFT . . . . . Discrete Fourier Transform

DNN . . . . . Deep Neural Network

DPCM . . . . . Differential Pulse Code Modulation

DRDNN . . . . . Deep Recurrent Denoising Neural Network

DTW . . . . . Dynamic Time Warping

DWT . . . . . Discrete Wavelet Transform

ECG . . . . . ElectroCardioGram

EEG . . . . . ElectroEncephaloGram

EMG . . . . . ElectroMyoGraphy

EWS . . . . . Early Warning Score

FCN . . . . . Fully Convolutional Network

FFNN . . . . . Feed Forward Neural Network

FFT . . . . . Fast Fourier Transform

GAN . . . . . Generative Adversarial Network

GRU . . . . . Gated Recurrent Unit

GSR . . . . . Galvanic Skin Response

HPC . . . . . High Performance Computing

HR . . . . . Heart Rate

ICD . . . . . Implantable Cardioverter Defibrillator

IoT . . . . . Internet of Things

LSTM . . . . . Long Short-Term Memory

LTC . . . . . Lightweight Temporal Compression

LZW . . . . . Lempel-Ziv-Welch

MAC . . . . . Medium Access Control

MCU . . . . . Microcontroller

mHealth . . . . Mobile Health

ML . . . . . Machine Learning

MLP . . . . . Multi Layer Perceptron

MRI . . . . . Magnetic Resonance Imaging

MTS . . . . . Multivariate Time Series

NN . . . . . Neural Network

PCA . . . . . Principal Component Analysis

PLA . . . . . Piecewise Linear Approximations

PPG . . . . . PhotoPlethysmoGram

ResNet . . . . . Residual Network

RFID . . . . . Radio Frequency IDentification

RLE . . . . . Run Length Encoding

RMSE . . . . . Root Mean Square Error

RNN . . . . . Recurrent Neural Network

ROC . . . . . Receiver Operating Characteristic

RR . . . . . Respiration Rate

SCL . . . . . Skin Conductance Level

SCR . . . . . Skin Conductance Response

ST . . . . . Shapelet Transform

SVM . . . . . Support Vector Machine

SZ . . . . . Squeeze

TSC . . . . . Time Series Classification

WBSN . . . . . Wireless Body Sensor Network

WSN . . . . . Wireless Sensor Network

I

DISSERTATION INTRODUCTION

1

INTRODUCTION

This thesis focuses on advancing the state of the art of IoT-healthcare, with a specific focus on time series data, energy-efficient data collection and the application of deep learning methods to solve various challenges. The presented research was carried out in the Department of Informatics and Complex Systems (DISC in French) of FEMTO-ST laboratory. Throughout the remainder of this thesis, the writer will be referred to as "we", rather than "I". This is because this thesis presents research performed in a collaborative setting in Belfort, France, as part of the AND research team.

This chapter provides an introduction to the work done in this thesis. It addresses the general context and the considered use case, then presents briefly the contributions of this thesis.

1.1/ INTRODUCTION TO IOT-HEALTHCARE

An Internet of Things (IoT) is a self-configuring, complex, and adaptive network connecting recognizable "things" to the internet. The "things" have capabilities for sensing and potential programmability. They are defined by information such as the identity, status, or location and offer services, with or without human intervention, through data collection, communication, and ability to take actions [1]. IoT applications are boundless, and the emergence of edge computing has enabled more optimized data processing, real-time analytics, and the use of artificial intelligence in many applications such as the energy and smart grid industries, connected vehicles and transportation, manufacturing, wearables, implantations, and medical devices [185].

A great deal of work has been done recently with IoT in the healthcare sector, and the benefits of IoT for healthcare are significant. One evolving role of IoT is to be able to remotely track patient health [179]. IoT-based services like telecare and mobile health, which allow medical well-being, prevention, tracking, diagnosis, and early symptom detection to be provided effectively through digital media, are expected to generate about $1.1-$2.5

trillion in annual growth for the global economy by 2025 [82]. Various IoT devices such as sensors and wearables may be used to capture physiological and behavioral data such as ElectroCardioGram (ECG), heart rate, blood pressure, weight, motion, etc. The measurements obtained can be transmitted over Bluetooth to the patient's phone. The phone then sends this data to a cloud storage database, along with patient information about the incident and symptoms. The data can then be analyzed and interpreted in order to extract the relevant information and notify a doctor in case the patient is at risk.

In view of the exponential increase in the number of internet connected devices and the immense amount of data generated by them, the use of artificial intelligence (AI) analytical and predictive techniques and especially deep learning techniques has become indispensable, particularly in healthcare analytics. The use of deep learning methods has produced unprecedented results in the fields of speech and face recognition, language processing, and object detection recently [147]. While the IoT-Healthcare-AI partnership has potential promise, major obstacles and challenges include reliable inter-device communication, efficient data collection, patient privacy and security, data cleaning, data classification, and user friendliness [143].

The work done in this thesis addresses numerous issues in an IoT healthcare framework, starting with data collection using sensors and wearables that involve data acquisition, processing, and transmission to data analytics.The data studied in this work correspond to physiological signals, i.e. time series.

The purpose of this thesis is therefore to suggest solutions to numerous problems and challenges that exist in IoT healthcare applications. The contributions concentrate specifically on the data collection process, in which efficient data transmission and energy-conservation are tackled, and the data analytics task, in which various deep learning approaches are presented to solve different challenges. Given the broad range of healthcare applications, the use case of human stress detection has been considered in this work.

1.2/ USE CASE: PHYSIOLOGICAL STRESS DETECTION

Latest technological advancements in the Wireless Body Sensor Network (WBSN), ubiquitous computing, and artificial intelligence provide new approaches and tools for affective computing.

Affective computing is an active research field attributed to Dr. Rosalind Picard of the Massachusetts Institute of Technology (MIT) that enables machines to identify and replicate human affect. To do this, different types of sensors or devices are required to collect data on the state of mind of the user, and then machines could use different machine

learning classification techniques to process and extract relevant patterns and then analyze these data.

The first basic affective states to be deduced are anger, disgust, fear, happiness, sadness, and surprise. In addition to these simple emotions, other complicated affects, such as human stress, can be deduced.

Stress is one of the major factors that negatively affect the performance of employees in the workplace or in complex activities requiring multiple cognitive abilities, such as driving. A recent study in Nigeria [221] has shown that factors such as poor control over the working environment and the management structure can contribute to job stress, resulting in lower performance and productivity of employees, a decrease in organizational reputation and a loss of skilled workers. According to the opinion survey "The Workforce View in Europe 2018" conducted by the private human resources company Automatic Data Processing [211], 18% of European employees say they are under stress every day at work. According to a recent study by Korn Ferry, "Workplace Stress Continues to Mount" [225], workplace stress has had a negative effect on workers' personal relationships and the quality of their sleep. In addition, the study found that stress can be transmitted, which suggests that supervisors who experience stress appear to transmit it to their employees through their own behaviour. The detrimental effect of stress does not only affect firms employees, but also people involved in different activities, such as driving. According to Vincente et al. [59], the driver's drowsiness, exhaustion, or stress induced by cognitive overload, tension, and prolonged driving hours can significantly disrupt the driver's ability to respond appropriately to relevant events.

For the reasons listed above, the study of stress has attracted a large number of researchers over the last few years. The demand for an automated stress detection system, which helps detect stress in its early stages and avoids burn-out in the case of office workers or accidents in the case of drivers, has increased. This dissertation considers the stress recognition use case where physiological signals are collected from IoT devices/sensors and sent to the edge/cloud for processing.

1.3/ Main Contributions of this Dissertation

The main contributions in this dissertation fall within the aforementioned phases of an IoT-healthcare application, namely data collection and data analysis and processing. The main contributions can be summarized as follows:

1. First, we propose an energy efficient data reduction scheme for IoT-Edge applications. The proposed method is based on an error-bounded lossy compressor designed for high performance computing applications that produce large amounts

of data during the execution, namely $Squeeze$ (SZ). The compression algorithm was adapted to fit on a Polar M600 wearable and resource-constrained devices such as ESP Wroom 32 microcontrollers. The performance of the proposed compressor was tested on medical multivariate time series. Furthermore, we considered the use case of drivers stress recognition and studied the impact of lossy data compression on the analysis, exploit, and classification of medical data. The results showed that the information extracted from compressed data was valid, and the classification accuracy obtained from training the model on features extracted from compressed data did not decrease.

2. Second, we investigate the effect of lossy compression techniques on the time series classification task using deep neural networks. The second contribution is a continuation of the first one, in which the proposed lossy SZ technique is combined with the lifting implementation of the discrete wavelet transform and compared with other compression techniques such as the Compressed Sensing. In order to study the impact of time series compression on the classification task, four newly proposed deep neural networks for time series classification were implemented. The impact of the various compression techniques on the task of time series classification with deep neural networks was studied, and the result showed that the proposed approach achieves an optimal trade-off between data compression and data quality.

3. The first two studies on the impact of lossy data compression on the classification task show that the quality of the data is highly degraded after a certain compression ratio and thus affects the output of the classifiers and data analysis, which is important in healthcare applications where the quality of medical data is of great importance. This issue is addressed by the third contribution of this thesis, in which a deep Autoencoder model is proposed to enhance the quality of the time series received at the edge/cloud before processing. The relevance of this contribution is that it allows the use of high compression ratio compressors at the sensor level to further minimize the amount of data transmitted, even if at the expense of further data distortion, since a sufficiently trained deep learning model can repair the received distorted data from the IoT devices.

4. In the fourth contribution, we tackle the issue of medical signals filtering. In this study, we consider the case of PhotoPlethysmoGram (PPG) signal that have potential use the in IoT-healthcare applications and specifically in stress recognition. Most wearable devices on the market today, are capable of collecting PPG data and enabling the measurement of important features such as heart rate, respiration rate, and blood pressure in addition to detecting irregular pulses and cardiovascular diseases. One major drawback of PPG data is its high sensitivity to motion, resulting in

distorted signal and meaningless data zones. We propose a neural network-based filtering method to remove corrupted zones from the collected PPG data in an unsupervised manner. We also propose a PPG data summarization and augmentation strategy which optimizes the network performance.

5. Finally, we tackle the multivariate time series classification task. Nowadays, most Internet of Things (IoT) devices collect multiple features and produce multivariate time series. In an IoT application, the mining and classification of the collected data have become crucial tasks. This study examines the use of the DenseNet architecture for the classification of multivariate time series, originally proposed for computer vision applications. More specifically, we are proposing a hybrid LSTM-DenseNet model that can achieve and surpass the performance of the state-of-the-art models in many situations, based on the results obtained from various experiments on 15 benchmark datasets. Thus, we suggest the 1D DenseNet as a potential tool to be considered by machine learning engineers and data scientists for IoT time series classification task.

1.4/ DISSERTATION OUTLINE

The rest of this dissertation is organized as follow: Chapter 2 discusses the scientific background of IoT in healthcare. Chapter 3 presents the wireless body sensor network and consumer health wearables technologies. Chapter 4 examines the different workflows of classic machine learning and deep learning approaches for IoT-healthcare. Chapter 5 presents the first contribution of this dissertation, namely an energy efficient approach for IoT data collection and analysis. Chapter 6 investigates the effect of lossy compression techniques on the time series classification task using deep neural networks. Chapter 7 resolves the limitations of the contribution presented in Chapter 5 by offering deep learning solutions to two different problems encountered in IoT-healthcare. Chapter 8 addresses the time series classification task and proposes a novel deep learning model that is able to achieve the performance of the state-of-the-art models and surpass them in many situations. Chapter 9 concludes the work that has been done in this thesis.

II

IoT in Healthcare: From data collection to artificial intelligence

This part examines the scientific background of IoT in healthcare by discussing the recent advances in this area and the numerous challenges faced by healthcare applications. Then we present the two tasks which are addressed in this thesis: data collection and data processing. The first chapter of this section deals with the healthcare applications and their requirements. The second chapter discusses Wireless Body Sensor Networks (WBSNs), wearables, physiological data, and energy-efficient data collection. The third chapter discusses the principles of IoT and artificial intelligence (AI).

2

IoT-Healthcare: Applications and Challenges

The continued development of IoT and the public adoption of wearable devices and biosensors has provided new possibilities for personalized eHealth and services and expanded the interaction between technology and healthcare. This chapter presents some of the applications of IoT in healthcare, and then discusses the challenges and requirements that need to be addressed in these applications.

2.1/ Applications of IoT in healthcare

The applications of IoT in healthcare are diverse and their purpose can vary from continuous monitoring of physiological parameters and event detection to providing medical assistance and diagnosis. This section presents various forms of IoT applications in healthcare.

2.1.1/ Mobile Health

Mobile Health (mHealth) consists of the use of mobile telecommunications technology for health care delivery and for wellness assistance. mHealth may help patients in two ways: (1) allow them to self-diagnose their physical symptoms more effectively and accurately, and (2) improve monitoring, tracking, and communication of various information such as blood pressure, glucose levels, oxygen saturation, etc. Additionaly, it allows people with chronic medical conditions to better participate in their treatment [85].

Widespread adoption of mHealth technology will enhance user experience by potentially enabling better monitoring of chronic diseases and enabling caregivers to exploit the mHealth mobile/web applications with P2P video/audio features to assist and direct patients at any time and anywhere [67].

2.1.2/ AMBIENT ASSISTED LIVING AND ELDERLY MONITORING

Ambient Assisted Living (AAL) is a term that incorporates emerging technology with the social environment to improve quality of living. This is achieved by using various types of sensors, mobile devices, software applications, computers, and wireless networks for the tracking of personal healthcare.

The aging population is a global problem, full of complex challenges. IoT technology may be used to unobtrusively assess the physical activity of the elderly, thus helping to prevent moderate cognitive dysfunction, hypertension, atrial fibrillation, depressive symptoms and control the posture and body motility of the elderly and to detect the occurrence of stroke [180, 181].

IoT technology solves this issues by allowing for indoor positioning as well as real-time location-aware tracking of vital signs (e.g., heart rate, blood pressure) and environmental conditions (ambient temperature), behavioral analysis and risk identification [139, 178].

Another essential use of IoT in AAL is the control of medication. Prototype systems have been developed for medication control using RFID (Radio Frequency IDentification) technology and tested in a simulated full cycle: from medical prescription to pharmaceutical drug management [66]. Self-alarm system is yet another example of IoT use for medication. It is common for many clients to forget to take their medication, which is an essential component of treatment. IoT may be used as a solution by monitoring changes in the condition of medicinal bottles with weight sensors, or by introducing real time continuous activity recognition with motion sensors [129, 132].

2.1.3/ INDIVIDUALS WITH SPECIAL NEEDS

Whether cognitive, psychological, sensory, mental or motor, disability comes in several ways. Nevertheless, in addition to this complexity, the difficulty of performing such basic acts in daily life is a matter of significant concern for all disabled persons. To counter the effects of one or more shortcomings, new technologies have been developed and reinforced over time to make remarkable progress today. IoT technology has opened up many opportunities for change in the field of disability treatment. So it is clear that these health-connected objects are part of an ongoing process of increasing the comfort and autonomy of disabled people. Far from replacing any human assistance, however, these internet-connected devices allow the interaction to be effectively sustained with family and skilled carers.

IoT devices can offer this disadvantaged population a good deal of support and dramatically enhance their lives through automated, timely, and reliable technologies. Smartwatches, for example, are used for patients with speech disabilities to improve their

speech functions in remote environments [115, 144]. In smart cities, IoT systems are designed to increase connectivity for wheelchair users who encounter mobility based daily challenges [107]. IoT may also help people collect information remotely about their special needs in their secure settings [145].

2.1.4/ Smart Medical Implants

IoT-healthcare also introduces very advanced, robust implantable medical devices intended to enhance or restore human body functions. Examples of some of these electronic devices include Deep Brain Stimulation (DBS) systems that use guided electrical pulses to activate brain areas in order to alleviate the involuntary movements triggered by neurological conditions such as Parkinson's disease [38, 40].

The cardiac pacemaker and the Implantable Cardioverter Defibrillator (ICD) are other examples [18]. One of the key signs for an ICD is sudden cardiac death, which affects about 100,000 people annually in the UK, showing the number of patients that may profit from this device [10]. Many implantable devices commonly used in medical practice include chronic pain implantable drug delivery systems [12], and high frequency brain (thalamic) stimulation for neurological disorders such as Parkinson's disease [8] and refractory epilepsy [11].

2.1.5/ Early Warning Score (EWS) and anomaly detection

The use of an Early Warning Score (EWS) system has long been encouraged by acute care teams to allow for a more auspicious response and assessment of critically ill patients [166]. An EWS system is based on a straightforward scoring system that assigns a score to each physiological measurement monitored from acute patients. The measures tracked are composed of six cardinal vital signs including temperature, respiratory rate, systolic blood pressure, pulse rate, oxygen saturation, and consciousness level. A typical healthy range is specified for each vital sign, and the assigned score indicates how significantly the calculated value differs from the norm. The farther outside the usual healthy range a calculated value is, the greater is the value of the assigned score. Combining all of the scores results in a composite showing the patients' cumulative risk of deterioration [149]. Note that EWS is an proven method commonly utilized by many hospitals worldwide. Several studies have shown that EWS can detect declining patients early in time, and thus can save lives and reduce patient mortality [110, 174].

Among the limitations of EWS is that it does not understand the full range of bio-signals like Cutaneous water/sweat and it may not be able to effectively identify certain anomalies that are not within the supervisor's expertise. Indeed, a precise method for detecting

anomalies involves continuous learning over time. The identification of anomalies is an significant aspect of health care analytics. In addition, anomalous zones are the segments of data in biomedical signals, or images that are of particular concern to medical staff. This field of study has gained tremendous interest and has a lot to bring to healthcare field. One of the primary goals will be to eliminate diagnostic error, increase early disease detection, and make healthcare accessible, and easy-to-use [162, 213]. Recently, numerous applications of anomaly detection in healthcare have been discussed, such as the detection of heart-related disorders using electrocardiogram signals [233], the detection of brain-related diseases and disorders such as epilepsy, Alzheimer's, and Parkinson's using electroencephalogram (EEG) signals [54, 130], and the detection of neuromuscular diseases using electromyography (EMG) [29].

2.1.6/ STRESS AND MENTAL-OVERLOAD DETECTION

Various sensors can now capture physiological and behavioral measurements that help identify human stress and mental overload such as accelerometer, gyroscope, blood pressure, electrocardiogram, electrodermal activity, and photoplethysmogram. In recent years, Koldijk et al have done extensive research on the development of automatic stress-related mental states detector for office employees by integrating various unobtrusive sensors [97, 151, 152]. The authors used body sensors to measure the physiological stress response from electrocardiogram (ECG) and electrodermal activity (galvanic skin response). In addition, other devices such as the webcam and the Kinect 3D camera were used to collect behavioral data such as body postures and facial expressions. In addition to numerous works dealing with well-being at work, others focused on identifying physiological arousal in driving activities by gathering heart rate, electrodermal activity, skin temperature, respiration rate, and motion data from drivers utilizing wrist-worn devices and body sensors [183, 184].

Wearables are actually most likely to be used by people who already have a balanced lifestyle and want to track their progress and enhance their physical fitness. The numerous research studies that examined stress-related mental states have shown that these wearables have immense potential to advance healthcare and affective computing applications by delivering accurate data that allows users to track physiological and behavioral improvement easily and unobtrusively. For example, the stress-related symptoms may be detected by using a commercial wearable wristband or smartwatch and a smartphone, based on the amount of physical activity, sleep duration, heart rate, and number of conversations [91, 123].

2.2/ REQUIREMENTS AND CHALLENGES OF IOT IN HEALTHCARE

Developing robust IoT-healthcare applications involves difficulties other than choosing the right and appropriate sensors and IoT devices. This section outlines some of the IoT-healthcare requirements and challenges to address before it could become a standard platform.

2.2.1/ DATA COLLECTION

Data collection is the first step in an IoT-healthcare application which needs to be carried out carefully to ensure its quality. The data collected must be reliable, valid, complete and comprehensive enough to help decision taking. Different physiological monitoring devices are now available and allow data collection to be of high quality [236, 237]. One difficulty, however, is the incorrect positioning of electrodes which would lead to meaningless measurements. Some sensors have established standards for electrode placement, such as ECG, and for other sensors, the correct placement is verified based on trials.

A further difficulty is finding a balance between the amount of data collected and the quality of the data. An example is the selection of the correct sampling frequency, given that a sampling frequency may be adequate for one type of signal and not adequate for others. The noise and artefacts present in the collected data are an important problem that is examined in this thesis using deep learning. This issue is generally solved by applying signal processing techniques to filter the noise using multiple filters such as Kalman filters, Median filters, and Wavelet decompositions. Due to the critical importance of the collected data for the decision taking and analysis process, significant attention is given to the data collection task in the following of this dissertation.

2.2.2/ BIG DATA

One of the greatest obstacles for IoT-healthcare applications is the large volume of data that is constantly collected. One way to prevent such a problem is to introduce data reduction techniques on sensor devices to prevent large volumes of data being transmitted and processed. Nonetheless, the energy consumption, battery life constraint, storage requirements and algorithm complexity are specific concerns that need to be addressed. In the state of the art, different methods were used to minimize the amount of data transmitted, such as adaptive sampling [193], data compression [192, 194], data prediction [215, 216], and dimensionality reduction [95].

Data segmentation is another problem when it comes to physiological and behavioral time series data. The use of sliding window strategies has been advised to prevent incorrect

selection of the segmentation windows which can lead to wrong decision taking [89]. Feature selection is another important problem when it comes to datasets with a high number of redundant information, which can make processing computationally expensive and decrease the classification accuracy. Various methods, such as the use of variance, correlation, forward selection, backward selection, and tree models, were tested [94].

Many data collection strategies may also be regarded and tested for different types of applications in healthcare. For example, it could be more effective to track only part of the data on an ongoing basis and activate the rest of the data monitoring when unusual events are observed, which may result in far less data collection without missing the important events.

2.2.3/ Unobtrusiveness and ubiquity

A key element in an IoT-Healthcare application is acceptance from the subjects. For example, relying on sophisticated sensors such as eye tracker to detect office workers stress lower the likelihood of such application being effective in a real setting. Minimal sensor configuration and unobtrusiveness are important criteria for the success of certain healthcare applications [100, 151].

With the developments in smart offices, behavioral data can now be obtained without distracting the user by incorporating sensors into seats, doors or simply by using computer-installed monitoring software (i.e. mouse and keyboard dynamics). Wearable technology and ubiquitous computing represent a new healthcare frontier by allowing for passive tracking of vital signs and the evolution of diseases such as cardiovascular, renal, or respiratory diseases, diabetes, and even cancer. Integrating sensors into consumer smart watches, bracelets, and other unobtrusive devices allows for the continuous and remote collection of vital data [158].

Another significant challenge is to provide the front-end devices (i.e. sensors, smartphones, tablets) with an easy-to-use interface. When designing and distributing IoT medical devices to patients who have little expertise in using high-tech technologies such as the elderly population, the user-friendliness and limited professional intervention are essential to the success of the healthcare application.

2.2.4/ Data integration

In many healthcare applications, the data collected typically comes from various sensors and devices, with various sampling frequencies. The integration of all such data constitutes a major challenge for these applications. Besides, the datasets do not have the same dimensions. For example, physiological univariate time series can be represented

with arrays, while multivariate time series and images can be represented with a matrix or tensors. One approach to solve these difficulties is to apply features extraction techniques and to work on the extracted features rather than on the raw data.

Missing measurements when dealing with data from many sources is another problem to be addressed. This prevents multimodal data from being compared at the same time. However, potential frameworks for data fusion may allow this problem to be resolved [7, 32]. Another challenge is that data from various sources may be inconsistent. A voting system could be used to solve this issue [52]. Another problem in data fusion is at what point to integrate the data into the recognition system. The data can be analyzed separately and fused into a final decision step or processed and merged sequentially. The reader is directed to [76, 120] for more information about the existing problems in data fusion.

2.2.5/ INTEROPERABILITY AND EFFICIENCY

Interoperability can be seen as a use of a system for another purpose. An example is the use of a proposed system for stress detection for continuous monitoring of the wellbeing. This can be accomplished by gathering data that can be feasible for different purposes and adopting existing physiological data coding and storage requirements that allow data to be shared between various medical devices and applications [28].

Efficiency is a key aspect of the wide use of the existing technology. It has been affirmed that several algorithms built in smart and academic environments to accomplish certain tasks, such as activity recognition, need improvement in order to become more reliable and effective in real life. When it comes to posing a danger to the safety of people and maintaining the credibility of a medical solution, the value of proper monitoring and meeting safety guidelines and certifications becomes increasingly important. For more details about government regulation of wireless medical devices the reader is referred to [81].

2.2.6/ PRIVACY AND SECURITY

IoT devices deployed in a healthcare system could pose a serious risk that might be exploited either to hurt end-users or to threaten their privacy. If IoT security is breached, information about end users can be reached through unauthorized authentications. In addition, this can pose risks for personal health. The Financial Times investigation found that 9 of the top 20 mobile healthcare applications were used to send data to parties interested in individuals mobile phone usage [84].

Different solutions are currently being tested to prevent the use of this knowledge for bad or non-ethical purposes. Data encryption [230, 239], patient authentication mecha-

nisms [229], and data anonymization [238] are among the safety precautions that could be used. The awareness and autonomy of the patients being monitored must also be ensured.

2.3/ CONCLUSION

Although the demand for medical care has soared in recent years, the conventional clinic-centered model of treatment requiring patients to see a specialist when sick is still the standard. This chapter presented an overview of how the progress in IoT help create significant developments in healthcare, by addressing various potential healthcare IoT applications. Additionally, there are many research issues that need to be solved carefully before it can become feasible for mainstream deployment. These issues were examined in the second part of this chapter and different solutions from the state of the art were presented.

3

Energy-efficient Data Collection

This chapter presents the wireless body sensor network and consumer health wearables technologies. Then discuss the key components of a biosensor node that consumes energy and introduces various energy-efficiency mechanisms used for energy conservation in the state-of-the-art. Finally, the data compression approach for data reduction is discussed and the main categories of data compression techniques are presented.

3.1/ Wireless Body Sensor Network and Consumer Wearables

The two promising technologies, namely the Wireless Body Sensor Network (WBSN) and consumer health wearables were targeted toward universal healthcare delivery [90, 154]. Nevertheless, the increased access to mobile devices, sensors and wireless networking solutions encourages the design and creation of pervasive assistance solutions that incorporate wearable devices and smart ecosystems to support people with severe disabilities, encourage diabetes detection and recognise emergency situations that may occur.

A Wireless Body Sensor Network (WBSN) consists of sensor nodes connected to a patient's body, or wearable devices worn by the patient that allow measurement of physiological parameters such as heart rate, respiration rate, ElectroCardioGram (ECG), PhotoPlethysmoGram (PPG), blood pressure, body temperature, electroencephalogram (EEG), and many others. For example, heart rate can be measured with an oximeter inserted into a ring or a smartwatch with a PPG sensor, muscle activity with an electromyographic sensor placed in clothing, stress with an electrodermal sensor integrated into a bracelet, and physical activity or sleep quality with a watch's accelerometer. Furthermore, the most fertile phase of a female can be identified with precise body temperature monitoring [105], whereas a limited number of non-gelled EEG electrodes may be used to measure mental concentration rates [126]. Social contact rates (also considered to impact general well-being) can be tracked with Bluetooth or WiFi-enabled devices, using proximity detections

to others [39].

These biosensors and consumer wearables send the collected measurements periodically to a coordinator, which is usually a portable device or smartphone, for fusion or aggregation. Meanwhile, for most consumer wearables and sensors, a coordinator is still necessary to process incoming data, but it is plausible that all processing capability would be self-contained in the coming years. After processing the data, the coordinator forwards them to the sink node or the base station [77, 168] as illustrated in Figure 3.1.

Although the WBSN platforms aim to provide and support a number of innovative and interesting applications [53, 156], there are many technical challenges that lie ahead such as the need for better sensor design, context-awareness, secure data transfer, and longer battery lifetime. Energy consumption is one of the key considerations for WBSNs since it not only determines the size of the battery required but also the duration that a biosensor can be left in situ [30]. The constrained sizes of biosensor nodes imply that they are designed with limited battery, storage, and processing. Therefore, there is a strong need for efficiency in data collection, analysis, storage, and communication.

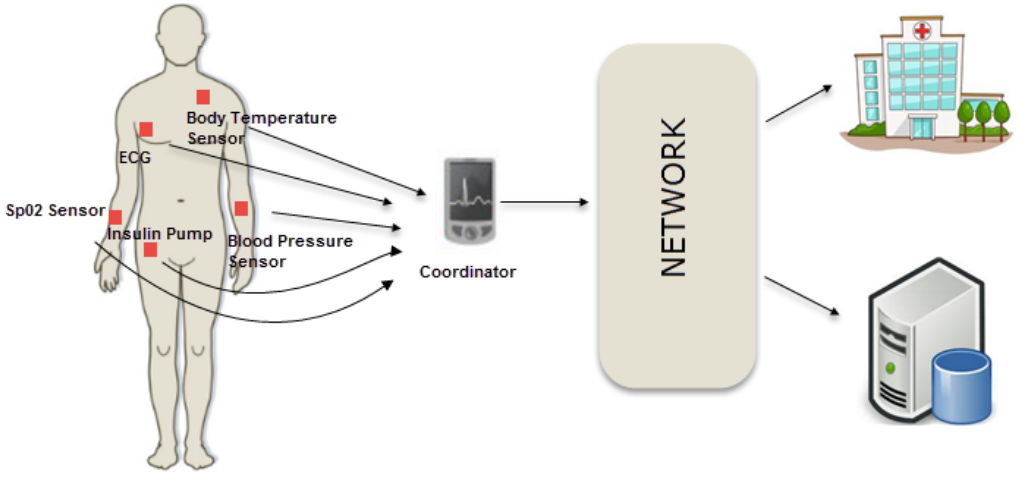


Figure 3.1: Wireless Body Sensor Network architecture

3.2/ BIOSENSORS AND ENERGY CONSUMPTION AT NODE-LEVEL

Biosensors are small, lightweight, low-power, resource-limited and smart sensor nodes that sense, process and transmit physiological parameters. Table 3.1 summarizes the most used sensors in medical applications, along with a short description of their functions and placement [77].

The four main constituents of a biosensor node that consume energy are the Microcontroller (MCU), Transceiver, Memory, and Sensor units. The MCU consumes energy for processing the data, executing code, and controlling tasks. The transceiver consumes

Table 3.1: Commonly used sensors in body sensor networks

Sensors	Function	Sampling Frequency	Placement
Accelerometer	Obtaining acceleration on each spatial axis of three-dimensional space.	High	Wearable
Artificial cochlea	Converting voice signal into electric pulse and sending it to implanted electrodes in ears, generating auditory sensation by stimulating acoustic nerves.	High	Implantable
Artificial retina	Receiving pictures captured by external camera and converting them to electric pulse signals, which are used to stimulate optic nerves to generate visual sensations.	High	Implantable
Blood-pressure sensor	Measuring the peak pressure of systolic and the minimum pressure of diastolic.	Low	Wearable
Camera pill	Detecting gastrointestinal tract by wireless endoscope technique.	High	Implantable
Carbon dioxide sensor	Measuring the content of carbon dioxide from mixed gas by infrared technique.	Low/Very low	Wearable
ECG/EEG/EMG sensor	Measuring voltage difference between two electrodes which are placed on surface of body.	High	Wearable
Gyroscope	Measuring angular velocity of rotating object according to principle of angular momentum conservation.	High	Wearable
Humidity sensor	Measuring humidity according to the changes of resistivity and capacitance caused by humidity changes.	Very low	Wearable
Blood oxygen saturation sensor	Measuring blood oxygen saturation by absorption ratio of red and infrared light passing through a thin part of body.	Low	Wearable
Pressure sensor	Measuring pressure value according to the piezoelectric effect of dielectric medium.	High	Wearable/ Surrounding
Respiration sensor	Obtaining respiration parameters indirectly by detecting the expansion and contraction of chest or abdomen.	High	Wearable
Temperature sensor	Measuring temperature according to the changes of materials physical properties.	Very low	Wearable
Visual sensor	Capturing features of subject, including length, count, location, and area.	High/Low	Wearable/ Surrounding
Photoplethysmogram (PPG)	Inexpensive optical measurement technique that measures changes in blood volume in the bed of tissue and is also used to track heart rate.	High	Wearable

energy for transmitting and receiving data. The energy consumption of the memory depends on the number of memory read and write, the number of stored bits, and the duration of storage. The Energy consumption of a sensing unit in an active state depends on the sensor radius, the data generation rate, and the number of generated bits.

The batteries of energy-constrained wireless sensor nodes are quickly drained with the ongoing tasks of sensing, processing, and transmitting and their regular replacement is not preferred, particularly as we want to motivate people to embrace this technology. Therefore, energy conservation is crucial in the data collection phase of IoT-healthcare applications.

Various energy-efficient mechanisms were discussed in the literature to address node-level energy-consumption challenges. These mechanisms can be divided into three categories representing the node's energy-consuming activities to provide continuous health monitoring, namely sensing, communication and processing.

3.2.1/ ENERGY-EFFICIENT SENSING

To maintain their energy level, energy-efficient sensing strategies focus on reducing sensing time or rate of sensor nodes. As a result, the unit(s) of the sensor nodes including radio, CPU and sensor(s) are disabled or the sampling frequency is changed to minimize the energy required to accomplish their tasks. Authors in [35] found that the assumption that data acquisition and processing activities consume significantly less energy than communication does not hold in a number of practical situations where the power consumption of the sensing operation is similar or even higher than that of communication.

- **Sensor set selection:** In this category, the techniques concentrate on achieving good balance between the number of enabled sensors and classification accuracy. Their ultimate goal is to optimize lifetime of a WBSN while maintaining good detection efficiency. The choice of the sensor set can take place before implementation or in real-time. Such an approach is used in applications for disease detection [60] and activity recognition [86].
- **Context-based pull:** Exploiting context correlation using approaches such as activity recognition is another approach to reducing energy consumption due to data collection [80]. Instead of following the general model where data is continuously transmitted from sensor nodes to coordinator, an asynchronous pull-based model is used. Thus, based on the defined user context, the coordinator requests data from the relevant sensor nodes. This reduces the energy consumption of continuous data collection without sacrificing application requirements.
- **Compressed sensing:** This technique exploits information rate within a specific signal and eliminates redundancy during the sampling phase, resulting in lower sampling rate [104]. When certain criteria are met, the signal can still be reliably retrieved, even when sampling at a sub-Nyquist rate. Compressed sensing and distributed compressed sensing can provide energy-efficient sensing in WBSNs.

3.2.2/ ENERGY-EFFICIENT COMMUNICATION

- **Radio Optimization:** The radio module is an important part which drains the battery from the sensor nodes. Researchers have optimized radio parameters such

as power transmission, antenna direction, transmission distance, transmission data rate, modulation schemes and coding to decrease the wireless transmission energy consumption [136, 155, 176].

- **Energy-efficient routing protocols:** Since the implementation of a WBSN affects the lifespan of the network, other researchers have provided energy-efficient routing protocols [88]. WBSNs encounter some of the same routing problems as general Wireless Sensor Networks (WSNs), but WBSN's specific specifications place some more restrictions that need to be tackled by routing mechanisms such as implanted bio-medical sensor nodes and human body heat sensitive organs. Current routing protocols can be categorized as: temperature-aware, QoS-aware, security-aware, cluster-based, cross-layered, and posture-based routing protocols based on operational mechanism [189].

- **Sleep/Wakeup schemes:** Medium access control (MAC) layer is essential for WBSN energy management. MAC protocol design would also ensure energy efficiency. Power management protocols can either be configured as independent sleep/wakeup protocols operating on top of MAC protocols that seek to adjust node operation by placing the radio in sleep mode to save energy, or strictly incorporated into the MAC protocol itself. Independent sleep/wakeup schemes can be classified into three categories: On-demand, Scheduled rendezvous, and Asynchronous [19, 36]. A significant importance of sleep/wakeup schemes is preventing excessive energy dissipation due to idle listening.

- **Data Reduction:** The main objective of implementing data reduction techniques on sensor nodes is to get rid of unneeded and redundant samples, thus reducing the amount of data to be transmitted. The mostly used data reduction approaches can be classified as follows [36] [103]: (1) In-network processing: This technique consists of performing data aggregation to reduce the amount of data to be forwarded to the coordinator, sink, or other nodes. For example, a node can only transmit the average, minimum, or maximum of the sampled data. (2) Data prediction: This technique consists of building a model describing the evolution of the data, and able to predict the sensed values within a certain margin of error. This model resides both at the sensor nodes and the coordinator/base-station. If the predicted values satisfy the accuracy needed, there is no need to transmit the data. However, when the prediction is not accurate enough, the sensed values must be transmitted in order to update the model. (3) Data compression: This technique consists of encoding the data in a way that the number of bits required to represent the compressed data is less than the number of bits required to represent the original data. Thus, reducing the number of packet transmissions.

3.2.3/ ENERGY-EFFICIENT PROCESSING

Healthcare applications like activity recognition and disease detection involve efficient, complex techniques to interpret and classify the signals. Some of the current research aimed at reducing processing energy consumption by using feature selection, adaptive classification selection, and energy efficient machine learning.

- **Feature selection:** Selecting features affects classifier efficiency. On the one hand, using multiple apps improves precision, but consumes a lot of energy and requires a lot of computation. By comparison, using a reduced number of features decreases computation and energy consumption at a lower accuracy rate. Therefore, a trade-off between delay/computation and accuracy should be considered [94, 173].
- **Adaptive classifier selection:** There are different types of classifiers that vary in terms of delay, memory and processing requirements. Therefore, some researchers have proposed in the context of energy-aware processing to adapt the classifier selection to the context of the individual being monitored, the battery capacity of the sensor nodes, available resources such as CPU load, available memory and the requirements of the application [75, 102, 173].
- **Energy efficient machine learning:** Since artificial intelligence and other types of cognitive computing keep developing, several researchers focused on exploring energy efficient machine learning techniques and architectures, particularly for edge applications and systems. Performance alone is never adequate for these resource limited environments, requiring system designers to carefully balance performance with power, energy and area. Different studies focused on topics such as neural network architectures for resource restricted devices [212] and comparisons of neural networks, based on Power, Performance and Area (PPA) [160].

3.3/ PRACTICAL DATA COMPRESSION IN IOT APPLICATIONS

One of the issues that this thesis addresses is the energy efficiency in IoT-healthcare applications. As seen in the previous section, different approaches to energy conservation could be adopted. The work in this thesis focuses on the data reduction approach, and particularly on data compression. Data compression can be considered as one of the direct methods that improves the energy efficiency, especially the energy related to the wireless transmission. The energy consumption of radio communication is directly proportional to the number of bits of data, that is, data traffic, transmitted within the network. Therefore, reducing the number of bits to be transmitted by compressing the data can increase the network lifetime.

Several compression techniques have been proposed in the literature to tackle the energy consumption problem. These techniques can be classified into three categories: sampling compression, data compression, and communication compression [83]. Sampling compression focuses on reducing the number of sensing operations while keeping the loss of information within an acceptable margin. Data compression, where the work of this thesis belongs, converts an input data stream into another data stream that can be represented with less number of bits. Communication compression focuses on reducing the number of packet transmissions and receptions. Typically, these three compression methods have a relationship. For example, a reduced number of samples helps minimize data length (data compression), which in effect decreases the number of transmissions.

A data compression technique may be lossless or lossy. The lossy compression method does not return the data to its original form after decompression. On the other hand, lossless compression restores and reconstitutes the data in its original form after decompression. A large number of compression techniques have been proposed in the literature. The following sections summarize text-based compression, data aggregation, transform-based compression, digital compressed sensing, predictive coding, and adaptive modeling.

3.3.1/ TEXT-BASED COMPRESSION

The text-based compression techniques view the collected data as a sequence of characters. Research works, which address the use of dictionary/text-based compression in WSNs are few in number. One of the popular lossless compression scheme for text data is the dictionary-based Lempel-Ziv-Welch (LZW) algorithm. An adaptive version of LZW algorithm, called S-LZW, has been proposed in [27]. S-LZW treats the collected data as strings and divides the strings into fixed-size blocks, with each being compressed using the LZW algorithm. Moreover, Huffman encoding and delta encoding have been utilized in the context of WBSN in [37]. The results revealed the superiority of dynamic delta encoding for WBSNs.

3.3.2/ DATA AGGREGATION

Data aggregation is any process in which information is gathered and expressed in a summary form. It is considered the simplest in-network processing technique for data and communication compression in WBSNs. Some aggregation techniques are based on aggregation function such as average, minimum, and maximum. However, aggregation techniques are rarely used in the context of WBSN because the original data cannot be reproduced at the coordinator or sink node.

3.3.3/ PREDICTIVE CODING

Predictive coding technique predicts future observations at the coordinator or sink node by using the inherent temporal correlation between consecutive readings at an individual sensor based on the statistical model and recent measurements [216]. The predictive coding techniques can use parametric modeling or non-parametric modeling, based on the nature of the sensor data. For parametric modeling, the statistical parameters such as mean and variance need to be learnt. Non-parametric modeling, on the other hand, uses regression to represent sensor data, and requires very little previous knowledge.

This technique thus decreases the number of communications between the source nodes and the sink, since nodes only send updates to the sink when new data arrives, or the difference between the expected value of the model and the observed value surpasses a threshold. Predictive coding, however, has a large overhead of learning data statistics [70, 83].

3.3.4/ TRANSFORM-BASED COMPRESSION

The transform-based techniques decompose a source output based on transform theories into coefficients that are then coded according to the individual characteristics [70]. These techniques are usually used for image and video signals.

Transformation techniques involve preprocessing the input signal by means of a linear orthogonal transformation and properly encoding the transformed output (coefficients) and reducing the amount of data needed to adequately represent the original signal. Upon signal reconstruction, an inverse transformation is performed and the original signal is recovered with a certain degree of error.

Several well-known approaches, such as Fourier transform, Karhunen-Loeve transform, cosine transform, and wavelets transform are adopted in certain types of WSN systems called wireless multimedia sensor networks [70]. Furthermore, transform-based compression can also be applied to sensed environmental data, such as temperature, humidity, light or even to vital signs data such as heart rate and respiration rate by modeling these data as an image map, and applying standard image compression methods to the map [83].

3.3.5/ DIGITAL COMPRESSED SENSING

The method has already been listed in the category of energy-efficient sensing. As this technique was proposed to handle the sensing task of the sensing unit by activating/deactivating the sensor, it was mostly used to contribute on the level of digital signal

processing [63, 79]. This is partly due to the shortcomings of the hardware that hinder the actual implementation.

Most of the compressed sensing-based data reduction techniques aim to reduce the energy consumption due to communication by reducing the volume of wirelessly transmitted data. Such methods minimize the sampling rate that was used to sample the digital signal, thus decreasing the amount of measurements to be transmitted. In other words, the digital compressed sensing technique impacts the processing and communication units instead of the sensing unit. The compressed sensing technique is further discussed in the third part of this dissertation.

3.3.6/ ADAPTIVE MODELING

In adaptive modeling approaches, some signal structure is iteratively updated over time, using linear, polynomial, or autoregressive methods to exploit the signal's correlation structure. In particular, the time series of inputs are gathered and processed according to transmission windows of N measurements each. The specified compression method is applied at the end of each time window, obtaining a series of model parameters which are being transmitted instead of the original data.

The Piecewise Linear Approximations (PLA) [228] and the Lightweight Temporal Compression (LTC) [232] are among the famous adaptive modelling techniques used in IoT applications. PLA's idea is to use a line segment sequence to represent an input time series with a bounded approximation error over specified time windows (of N samples). LTC is a simple lossy algorithm, just like the Run Length Encoding (RLE) in the sense that it tries to represent a long sequence of similar data with a single symbol by searching for linear trends.

3.4/ CONCLUSION

An overview of the biosensors being used in WBSNs has been given in this chapter. It has been shown that there are different types of biosensors based on wearability and signal type and that smart watches and other wearable devices are currently gaining a lot of attention due to their usability and potential value in healthcare applications.

In addition, this chapter covered the most common energy-efficient mechanisms used in WBSNs. They have been classified into three categories: sensing, processing and communication, depending on which energy consuming data collection task is primarily targeted. Since this thesis particularly addresses the data compression approach for data reduction, different types of practical data compression techniques have been presented.

4

Artificial Intelligence for IoT Healthcare

The second important task in IoT-healthcare addressed in this thesis after data collection is data processing and analytcs. The large volume of data generated by WBSNs and wearables requires the use of artificial intelligence analytical and predictive techniques such as machine learning and deep learning. This chapter introduces the use of artificial intelligence in IoT, then addresses the machine learning and deep learning approaches for IoT-healthcare. Furthermore, it presents the numerous challenges in health informatics that artificial intelligence technologies face.

4.1/ Introduction

Healthcare is entering a new phase in which the enormous amount of biomedical data is playing an increasingly important role. For example, personalized medicine aims to ensure that the right care is given to the right person at the right time by taking into consideration many elements of health data including variation in molecular characteristics, environment, and digital health records. The previous chapters have illustrated the important role IoT plays in healthcare and what potential this technology has.

IoT creates massive volumes of data. 90% of currently produced data are not even retrieved, and out of the 10% captured, the majority are time-dependent and lose their value within milliseconds. Continuous manual analysis of such data is both costly and inefficient. Artificial Intelligence (AI) tools provide a way to evaluate and gain insight from data smartly with minimal human interference.

AI is the technology that can be implemented in machines and allow them to perform tasks that demand human intelligence. These tasks involve seeing and identifying things, identifying sounds, understanding and interpreting language, and other related tasks. Machine learning (ML) is a subset of AI that comprises methods used to enable such

human-like tasks [169]. ML encompasses methods that can be used to achieve AI and gives machines the opportunity to learn without being explicitly programmed. This is done by including examples of how a task is done and how it should not be done, instead of relying on hardcoding rules.

With the growing role of data analytics and ML in health informatics, deep learning has emerged as a powerful method for machine learning in recent years, promising to transform the future of the healthcare sector [235]. Deep learning is a special tool in ML which simulates the biological structure of the human brain and works to execute human-like functions. This is achieved by constructing a network of neurons just as in the brain through the use of a computational approach such as artificial neural networks, which can solve problems at human-like performance or better.

4.2/ MACHINE LEARNING APPROACHES FOR IOT-HEALTHCARE

To teach machines how to solve a great number of problems on their own, various machine learning models need to be considered. The model needs to be fed with data; that is why, based on the input datasets, machine learning models are divided into four main groups: supervised learning, semi-supervised learning, unsupervised learning and reinforcement learning.

Machine learning techniques are able to automatically analyze and derive information from the data being collected, thereby promoting rigorous and more informed decision making. This section discusses some of the common medical decision-making frameworks and compares them to machine learning methods.

4.2.1/ CATEGORICAL AND TREE-LIKE REASONING

Tree-like reasoning requires branching into different potential actions when specific decision requirements are fulfilled. A medical question may be viewed as a tree when the problem is perceived to be an organized, arranged set of rules for making a decision. The root of the tree in this sort of reasoning reflects the beginning of the interaction with the patient. As the doctor gathers more details, he comes to various decision points where he can continue in more than one path. These paths may represent various medical tests or additional lines of questioning. The doctor makes decisions continuously and select the next path until reaching a terminal node where there are no paths remaining. The terminal node denotes a treatment plan that is definite [58].

Decision tree and random forest can be considered as the corresponding algorithms for machine learning to such problems. Decision trees developed in the 1990s and used

information theory concepts to optimize tree branching variables to improve classification precision. For decades, the standard decision tree algorithms have existed and modern variants such as random forests are among the most effective techniques available for supervised classification tasks. Such an approach has the advantage of being able to select the most important features and combine numerical and categorical data. However, when the problem size increases, the calculations exponentially grow.

4.2.2/ PROBABILISTIC REASONING

A second way to approach the patient encompasses initializing a patient's baseline probability of a disease and adjusting the likelihood of the disease with each new medical finding. One of the algorithms corresponding to probabilistic reasoning is the Naive Bayes classifier. Naive Bayes is a predictive modelling algorithm that is simple but remarkably powerful. Naive Bayes is a binary (two-class) and multiclass classification problem classification algorithm, which calculates the likelihood of each class being correct by applying Bayes' theorem. One of this approach's main drawbacks is that it does not tackle dependent variables. In the health informatics literature, probabilistic reasoning has been widely used to solve issues such as the identification of cardiovascular disease risk's level and the prediction of various dermatological conditions [101, 153]. Additionally, it has been used for public health applications such as air pollution prediction [210].

4.2.3/ COMPLEX CLINICAL REASONING

Many research focused on more complex issues such as detecting "alarming events" (e.g., a fall) that are typically handled in a discriminative manner, i.e. learning to differentiate an alarming situation from non-harmful situations based solely on the data previously monitored and finding dependency of annotations (ground truth). Discriminative (often called conditional) models can solve these challenges.

- **Logistic Regression:** Logistic regression is named for the method's core function, logistic function or sigmoid function. The most common task this approach solves in practice is binary classification. It has been used to diagnose moderate cognitive disability [21] and for tasks such as fall detection using cell phones [61].
- **Support Vector Machines (SVM):** SVM is a supervised machine learning model that works by finding the optimum hyperplane capable of separating two classes This plane is identified as the one with the greatest margin towards the nearest data point of each class. A kernel transformation can be implemented to help define the correct hyperplane. The SVM has been used for fall detection tasks by

wearable sensors [31] and for presence detection of visitors in a smart home environment [68].

- **Conditional Random Fields:** Conditional Random Field is a specific case of Markov Random field ideally suited for predicting tasks where contextual information or neighborhood condition influences the current prediction. It was used to track elderly activity and recognize accelerometer-based movement [56].
- **Boosting:** The idea of boosting came from the question of how to strengthen a slow learner. Gradient boosting is one of the most efficient predictive modeling techniques. XGBoost is a gradient-boosted decision trees developed for speed and performance, and used for various health-related tasks such as drowsiness detection and orthopedic auxiliary classification and prediction [198, 227].
- **Artificial Neural Networks (ANNs):** A neural network is modeled after the mammalian nervous system, where independent variables are connected to sequential layers of artificial ”neurons” that aggregate and sum weighted inputs before transmitting their nonlinearly transformed outputs to the next layer. In this way, data can move through multiple layers before eventually generating an output variable indicating the probability of an event. ANN has been widely used in health informatics for tasks such as detection of heart-related disorders, activity recognition, and diagnosis of diabetes [34, 65].

4.2.4/ Regression problems

Numerous works in the literature focused on solving regression problems where the objective is to estimate a continuous variable. Different techniques can be used for regression problems such as linear regression, support vector regression, decision tree regression, and random forest regression. Regression techniques have been used in health information technology to predict functional ability [44], estimate mortality risk in elderly population, or continuously predict vital signs such as blood pressure based on certain factors that are noninvasively measurable [57].

4.2.5/ Time series and sequential data

Time series consist of various types of physiological data obtained from WBSNs or wearables. The points in time are usually similarly spaced. Measurement frequency depends on the type of the data and its applications. To make sense of the collected physiological time series, stochastic signal analysis techniques in combination with conventional machine learning classifiers can be used for accurate classification and modeling. Various hand-engineering feature methods can be used to extract features from signals and

be able to accomplish numerous tasks such as classification of ECG signals, prediction of seizures from EEG signals, and identification of patients with neuropathy from EMG signals.

- **Fourier and Wavelet Transforms:** A Fourier Transform converts a signal into its frequency domain equivalent, and the Fast Fourier Transform (FFT) algorithm easily calculates these transformations by factorizing the Discrete Fourier Transform (DFT) matrix into a product of sparse variables, including frequency data such as magnitude, amplitude, phase, power density and other calculation results. Those extracted features may be used as classification features[1] [33]. Wavelet transform is another common time-frequency transformation method for features extraction. Wavelet transform is designed to tackle the non-stationary signal problem. It involves the representation of a time function in terms of simple, fixed building blocks, called wavelets. The Discrete Wavelet Transform (DWT) has been used as a feature extraction mechanism, as it has been shown to efficiently solve health-related classification problems [138].

- **Principal Component Analysis (PCA):** PCA is a statistical method that uses an orthogonal transformation to map a set of observations of possible correlated variables into a collection of values of linearly uncorrelated variables called principal components. Generally PCA is considered a technique for data reduction. A property of PCA is that the number of dimensions or key components in the transformed output can be chosen. The PCA was used in [48] to extract relevant features from the ECG signal to detect ischemic beats.In [167], the authors applied PCA to analyze physiological readings obtained from sensors in order to detect the occurrence of multivariate anomalies.

- **Canonical Correlation Analysis (CCA):** CCA connects two sets of variables by identifying linear combinations of maximally correlated variables. Using a small number of linear combinations, it can be used for data reduction to describe covariation between two sets of variables. Additionally, it can be use for data interpretation by identifying features that are vital to explain covariation between sets of variables. CCA has been used for numerous tasks in health informatics, such as extracting motion artifacts from physiological signals by computing the correlation coefficients [182].

[1] The following example illustrates how digital signal processing techniques such as Fourier transform can be used in stress detection tasks for features extraction: https://josephazar.net/blog/dspstressml1.php

4.2.6/ ANOMALY DETECTION

The problem of finding items or patterns in a dataset that do not correspond to other items or an anticipated pattern is called anomaly detection. Medical and public health administration are key applications for anomaly detection. There are three major types of anomaly detection techniques, based on the degree of available labels. A binary (abnormal and normal) dataset is given in supervised anomaly detection techniques, then a binary classifier is trained. The issue of unbalanced data resulting from having few data points with an abnormal label remains a challenge to address. Semi-supervised anomaly detection techniques involve a training set of only normal data points. Unsupervised anomaly detection methods deal with unlabeled dataset by assuming that most data points are normal. One-class support vector machine is an example of an anomaly detection technique that can be used in an unsupervised manner to solve various tasks including medical diagnosis [71].

4.3/ DEEP LEARNING APPROACHES FOR IOT-HEALTHCARE

Representation learning is a modern, rapidly evolving approach to machine learning in recent years that allows automatic features learning from raw data and to some extent solves the hand-designing features problem. Deep learning is a standard approach to representation learning. Unlike classic machine learning algorithms, in which features need to be given manually, deep learning techniques automatically extract important features after determining the features that are the most important for classification or clustering. Figure 4.1 illustrates the different workflows of classic machine learning and deep learning approaches. Every block in the deep learning workflow gradually extracts the input features and then processes data from the previous blocks, which have already been processed, extracting more abstract data features and thereby building up the hierarchical representation of data. Deep learning techniques can be divided into supervised learning techniques such as Convolutional Neural Network (CNN) [5] and Recurrent Neural Network (RNN) [2], and unsupervised or semi-supervised techniques such as deep Autoencoders [24], deep belief networks [23], and deep Boltzmann machines [41].

Recently, many advances in healthcare analytics have been made using deep learning. Deep learning methods have been applied to various areas and applications in health informatics where the performance of the classical approaches to machine learning is limited, for example in medical imaging and bioinformatics. Figure 4.2 summarizes the different applications of deep learning by areas in health informatics.

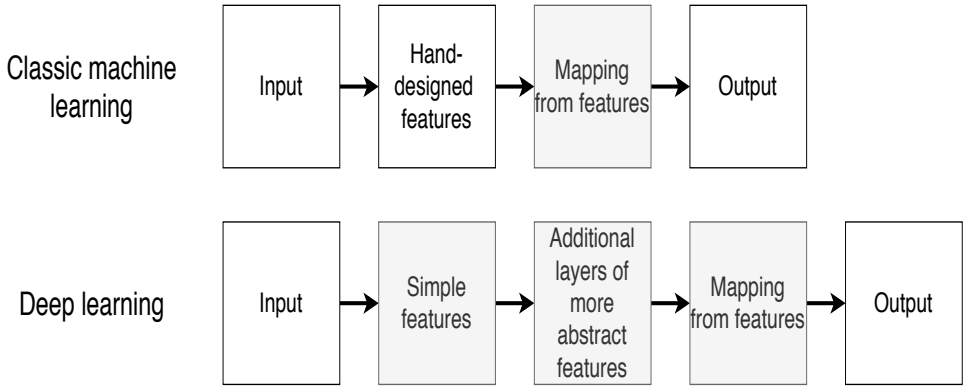


Figure 4.1: Flowcharts showing the different parts of classic machine learning and deep learning approaches. Shaded boxes indicate components that are able to learn from data.

4.3.1/ BIOINFORMATICS

Bioinformatics aims at researching and interpreting biological processes at a molecular level to predict or prevent disease. Deep learning applications in bioinformatics include cancer diagnosis, gene recognition, drug design, and DNA methylation where data input ranges from gene expression to sequences of DNA and molecular compounds. The vast volumes of data collected from recent biological experiments contributed to the creation of huge databases containing data on proteins, genes, and other genetic data [219]. In addition to big data-related technologies (e.g. cloud computing, IoT, Hadoop), deep learning's ability to abstract big, complex, and unstructured data provides an efficient way to analyze such heterogeneous data. The authors in [73] proposed a method for enhancing the diagnosis and classification of cancer from gene expression data using an using unsupervised deep leaning approach based on a deep Autoencoder. By collecting vast amounts of data from public sources, the authors in [127] investigated the use of massive multitask neural networks for drug discovery that synthesize information from several different biological sources. In [137], The authors proposed a computational method to predict DNA methylation states from the DNA sequence and incomplete methylation profiles in single cells based on deep neural networks.

4.3.2/ MEDICAL IMAGING

Deep learning techniques are used in medical image processing to efficiently and effectively extract information from medical images which are obtained using various imaging modalities such as magnetic resonance imaging (MRI). Such techniques provide essential functional and anatomical details about different body organs, and play a crucial role in detecting/locating and diagnosing anomalies. Deep learning applications for medical imaging range from improving poor medical images with convolutional autoen-

Bioinformatics	Medical Imaging	Medical Informatics	Public Healtth	Pervasive Sensing
Cancer diagnosis	3D brain reconstruction	Prediction of disease	Demographic info prediction	Human activity recognition
Gene classification	Brain tissues classification	Human behavior monitoring	Air pollutant prediction	Time series classification
Drug design	Organ segmentation	Temporal patterns discovery	Epidemic surveillance	Sign language recognition
DNA methylation	Tumour detection	Multi-modal data mining	Lifestyle diseases tracking	Anomaly detection

Figure 4.2: Summary of the different deep learning applications by areas in health informatics.

coders [146], to detecting complex patterns of disease such as tumor or cancer with CNN-based techniques, and segmenting tissues and organs in medical images using CNNs and RNNs [133].

4.3.3/ MEDICAL INFORMATICS

Medical informatics involves the study of massive aggregated data in healthcare environments with the goal of improving and developing support systems for clinical decision-making or analyzing medical data for both quality control and usability of healthcare services. Electronic health records contain vast quantities of patient data such as descriptions of medical history, diagnoses, medications, immunization records, radiology images and multivariate physiological time series such as ECG or EEG. Deep learning has proved to be an effective method to mining this big data, which may provide useful insight into disease prevention. The advantages of deep learning over classical machine learning is that deep learning methods can very well scale up with broad and distributed data and cover multimodal datasets including medical images, sensor data, and unstructured text reports [131]. In addition, deep learning methods such as RNNs can effectively resolve long-term dependency between clinical events and the diagnosis and treatment of diseases [122].

4.3.4/ PUBLIC HEALTH

Public health aims at preventing illness, prolonging life, predicting demographic info, and improving wellness by examining disease transmission such as pandemics, and social activities in relation to environmental factors. The data used in this area of health informatics are social media data, mobile phone metadata, and text messages. A deep recurrent neural network enhanced with a pre-trained autoencoder has been proposed in [157] to predict air pollution. In [125], the authors proposed a deep learning model based on Restricted Boltzmann Machine to model and predict human traits, such as physical activity and intensity, and help spread well-being and healthy behavior in a social network. In

addition, natural language processing and sentiment analysis can play a huge role in this area, such as relying on Twitter messages and Facebook posts to identify new mothers at postpartum depression risk.

4.3.5/ PERVASIVE SENSING

The research works done in this dissertation are mainly concerned with this section of health informatics. Pervasive sensors like wearable, implantable and ambient sensors enable continuous health and well-being monitoring. The data collected can consist of physiological time series obtained from wearable or implantable devices, images from sensor cameras, vital signs and food intake information. The applications of pervasive sensing are diverse, ranging from recognition of human activity [74], generic object recognition, and gesture recognition using convolutional neural networks, to detection of anomalies in vital signs and vast quantities of time series using techniques such as autoencoders, deep-belief networks, and recurrent neural networks.

4.4/ CHALLENGES OF ARTIFICIAL INTELLIGENCE IN HEALTHCARE

There are several obstacles to the application of artificial intelligence techniques in healthcare. Also, ethical issues raised by this technology are highly discussed and must be addressed. Finally, while obstacles imply a possibility of being overcome, this technology also has limitations, or aspects that are unlikely to be resolved soon.

4.4.1/ OBSTACLES

Healthcare has typically been slow to adopt emerging technologies, a challenge that needs to be overcome. Healthcare was characterized as a conservative sector, slow to accept change. For instance, there was initial resistance to using electronic blood pressure cuffs in hospitals. Analytics and artificial intelligence are definitely no exception; it is just another modern, innovative technology, and while industries such as automotive and manufacturing adopted it with little concern, healthcare is likely to be another case. Perhaps an important fundamental reason why doctors resist analytics and machine learning is fear of computers trying to “take over” or “replace” doctors. The fear that doctors have about being replaced by machines is a real possibility, albeit distant, ways must be found for doctors and artificial intelligence to work together rather than against each other.

Another explanation for theoretical skepticism is “hope versus hype” argument [64]. Buzz words like “big data” and “deep learning” often bear negative connotations due to their

hype. Some people assume over-inflated confidence in these fields. Specifically, analytics and machine learning skeptics argue that most large-data applications, while ”sounding cool”, rarely save lives or money.

4.4.2/ ETHICAL ISSUES

Ethics has always been part of assessing new technologies, including computer science in IoT-healthcare. The first ethical challenge is the inability to give human feelings and painlessness a value or number. A cost function is used by many machine learning models for training. What is the cost function in these cases? Will it be based on decreasing costs, and increasing outcomes and quality? or decreasing pain and emotional turmoil? Who determines the ratio to undertake these opposing goals? Another ethical issue concerns responsibility. If a machine learning model makes an incorrect judgment, who is responsible? Should it be the physician or the data science team that made the model? A third problem lies in patient privacy. Is using patient data to train models appropriate? Should this require consent or not? What data points should be used? Finally, there is also the bias problem. There is concern that estimating health outcomes may rely on things like race, gender, and age. This can result in patient discrimination.

4.4.3/ LIMITATIONS

Limitations are those aspects of healthcare analytics that a data scientist may face while designing applications for healthcare. An significant weakness of deep learning is the ability to clarify decisions. The decision-making method of a machine learning system should be transparent to the user. For example, when a certain health risk is automatically identified, an algorithm will provide a description of circumstances that prompted such a warning. For instance, graphic models and decision trees are more appropriate than black box neural networks algorithms, where mathematical reasoning is hard to explain.

Another big problem in healthcare analytics is the imbalanced classification issue. This issue is an example of a classification task where the distribution of samples is not equal across known classes. Distribution can range from a slight bias to an extreme disparity where there is one example for hundreds, or thousands of examples in the majority class. Imbalanced classifications present a challenge for the predictive models because most of the machine learning techniques used for classification assume an equivalent number of examples for each class. This results in low prediction performance, especially for minority classes that are extremely important in healthcare applications. Other limitations are the presence of noise in physiological time series due to sensors misplacement or sensitivity to motion. In addition, patient data collection is often costly and time-consuming for

doctors, it is important to have a classifier that can accurately identify a health threat with fairly low amount of training data about the patients.

4.5/ CONCLUSION

The different approaches for machine learning and deep learning for IoT-Healthcare have been discussed. While the multiple success of machine learning in different healthcare fields has been achieved, deep learning provides more potential success when it comes to medical imaging, text messages, and a vast amount of physiological data. A pipeline of layer by layer results in an efficient high-level abstraction of the raw data or images. This high level of abstraction allows an automated set of features, which would otherwise require human interaction for sophisticated feature extraction. Finally, the obstacles, ethical issues, and limitations facing the application of artificial intelligence technologies in healthcare are discussed. The emphasis of the following works presented in this dissertation is on applying deep learning to pervasive sensing, where physiological data is obtained from wearable or sensor devices and transmitted for analysis to the sink.

III

CONTRIBUTIONS

This part examines the main contributions in this thesis. As discussed in the previous chapters, the proposed approaches and methods are designed to solve different challenges in the data collection and data processing phases of an IoT-healthcare application.

5

IoT DATA COMPRESSION FOR EDGE MACHINE LEARNING

IoT-healthcare systems generate a huge and varied amount of data that need to be processed and responded to in a very short time. One of the major challenges is the high energy consumption due to the transmission of data to the cloud. Edge computing allows the workload to be offloaded from the cloud at a location closer to the source of data that need to be processed while saving time, improving privacy, and reducing network traffic. In this chapter, we propose an energy efficient approach for IoT data collection and analysis. First of all, we apply a fast error-bounded lossy compressor, namely *SZ* (SZ), on the collected data prior to transmission, that is considered to be the greatest consumer of energy in an IoT device. In a second phase, we rebuild the transmitted data on an edge node and process it using a supervised machine learning technique. To validate our approach, we consider the context of driving behavior monitoring in intelligent vehicle systems where vital signs data are collected from the driver using a Wireless Body Sensor Network (WBSN) and wearable devices and sent to an edge node for stress level detection. The proposed technique was implemented on a Polar M600 wearable, ESP Wroom 32, and WiFi LoRa 32 microcontrollers.

5.1/ INTRODUCTION

Wireless Sensor Networks (WSNs), Wireless Body Sensor Networks (WBSNs), and wearable devices make up the essential blocks of IoT architectures. Many of these smart objects, that are responsible for the collection, processing, and transmission of data, are still battery operated and resource constrained. The three major constituents of a smart object that consume energy are the microcontroller (MCU), transceiver, and sensor units. Among all tasks, it is well known that data transmission is the highest energy-consuming task in IoT nodes [36] [83]. An important step towards energy efficiency in IoT applications is the transfer of computational tasks from the cloud to the edge. In general, the

radio communication task between the IoT nodes and the edge consumes less energy than transmitting the data directly to the cloud over the cellular network [47, 161]. Equally important, reducing the amount of data to be transmitted to the edge can further increase the lifetime of the IoT nodes and save storage at the edge.

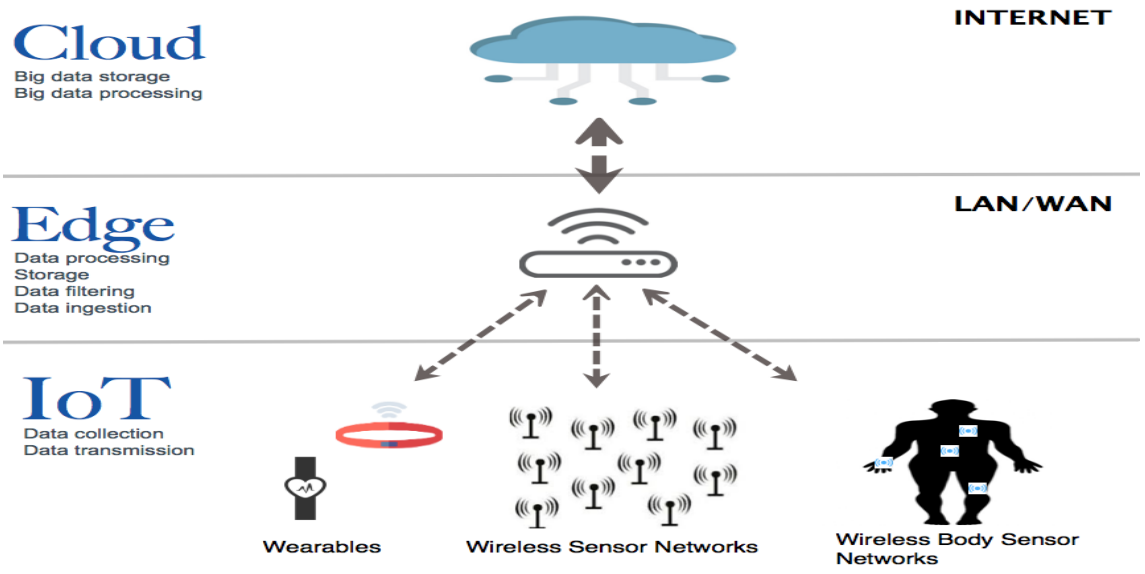


Figure 5.1: IoT network architecture

This chapter considers the IoT network architecture shown in Figure 5.1, where the data are collected from smart objects such as wearables and sensor devices and sent periodically to an edge node using short range communication protocols (e.g. WiFi, Bluetooth). The edge node is responsible for the processing, analyzing, filtering, storing, and sending the data to the cloud. To start, The energy conservation problem is tackled by proposing a fast error-bounded lossy data compression technique to be deployed on the IoT devices. The objective is to reduce the number of bits to be transmitted periodically to the edge node. Then, we study the effect of lossy compression on the performance of machine learning models deployed at the edge, and trained on reconstructed data with degraded quality as compared to the original data. To do so, we consider the driving behavior monitoring use case where physiological signals are collected from drivers and sent to the edge node in order to detect their stress level. The stated problem in this work can be formulated as follow: *To what extent does the loss of information due to lossy compression and energy conservation techniques deployed on the IoT nodes affect the analysis and processing of the data at the edge?*

5.2/ RELATED WORK

In this section, we first introduce the analysis of data in edge computing using machine and deep learning, and then we discuss data reduction in IoT applications.

5.2.1/ MACHINE AND DEEP LEARNING IN EDGE COMPUTING

The use of machine and deep learning techniques for data processing could help edge devices to be smarter, and improve privacy and bandwidth usage. In [209], the authors introduced deep learning for IoT into the edge computing environment and proposed an approach that optimizes network performance and increase user privacy. Machine and deep learning in edge computing can bring multiple improvements to the traditional approaches that rely on cloud computing by:

- Processing data using conventional machine learning techniques and transferring the results or necessary features extracted from raw sensor data.
- Deploying part of deep learning networks layers on the edge and transferring the extracted features whose size is smaller than that of the input data.
- Deploying neural networks on the edge with minimized size that maintain accuracy.
- Training the networks on the cloud and shipping the trained models to the edge.

The aforementioned approaches reduce the pressure on the network by reducing the size of the data to be transferred to the cloud. For instance, each layer in a deep learning network processes and scales down the size of the generated features from the previous layer. The more layers are deployed on the edge, the smaller is the size of the features to be transferred to the cloud and the more they are incomprehensible, hence, increasing the privacy.

Edge nodes are devices such as mobile phones, IoT gateways, and local PCs that have a limited processing capability as compared to cloud servers. The size of neural networks deployed on these devices should be reasonable. In [163], the authors showed that different lightweight libraries and algorithms can be deployed on edge nodes such as smartphones and enable real-time data analytics.

5.2.2/ IOT DATA REDUCTION

Different data reduction schemes for energy saving in IoT applications have been proposed in the state of the art. In [78, 174], the authors proposed data reduction approaches

based on adaptive sampling. These approaches work by studying the level of variance between the collected data over a certain time frame, and dynamically adjusting the sampling frequency of the devices. Adaptive sampling approaches work well in applications where the collected time series are stationary. In the case of quickly varying data, these approaches perform poorly. In [216], the authors proposed a data reduction mechanism based on dual prediction. The proposed mechanism works by building a model that describes the sensed phenomenon and deploying it on both the edge node and the IoT devices. The advantage of prediction approaches is that the model at the edge predicts the sensed measurement without requiring any radio communication unless the prediction error exceeds a predefined threshold. However, such prediction mechanisms suffer when it comes to devices like a high frequency motion sensor, where the data collection frequency is high and the data varies quickly.

Numerous aggregation approaches that take advantage of the temporal correlation in the collected data have been proposed [190, 203, 204]. In [203], the authors proposed a data filtering technique based on the Pearson coefficient metric. This method works by recursively dividing the dataset into two equal parts and aggregating the data based on the correlation between the subsets. In [190], the author proposed a data aggregation method for incoming data stream in IoT based monitoring systems. The proposed method is an approximation with 'extremums' technique that reduces the volume of data to be stored or transmitted. The results show that this method was able to achieve a compression of up to 10 times on temperature data.

5.2.3/ IoT DATA COMPRESSION

It is possible to classify data compression algorithms into two types: lossless compression and lossy compression. Lossless compression does not involve data loss, while lossy compression leads to some data loss. However, the maximum data reduction that can be reached by lossless compression is limited. Typically, when choosing the appropriate algorithm, the main property to check is the Compression Ratio (CR) factor, denoted in Equation 5.1:

$$CR = \frac{uncomp\_size}{comp\_size}, \tag{5.1}$$

where $uncomp\_size$ is the size of the original data, and $comp\_size$ is the size of the compressed data.

Lossless compression techniques work on a basis to eliminate data redundancies. Huffman coding, Arithmetic Coding, Run-length Encoding (RLE), dictionary-based general purpose techniques such as Lempel-Ziv compression, are the most commonly used loss-

less compression techniques. More recently, Sprintz has been proposed in [196] for IoT applications as an efficient high-ratio lossless compression technique for multivariate integer time series.

If some insignificant data can be omitted in data transmission, a compression technique is said to be lossy. Many IoT applications do not require accurate signal reconstruction and further analysis can be performed on signals that are compressed with lossy techniques. One major advantage of using lossy techniques over lossless techniques in IoT applications is that the lossy compression yields higher compression ratio and helps to reduce the amount of data transmitted. Lossy compression techniques can exist in time domain such as curve fitting, fractal resampling, box car and in frequency domain such as wavelet transform, Fourier transform, and Chebyshev compression.

In [192, 194], the authors proposed data compression techniques that take advantage of the temporal correlation in the collected data. The proposed techniques are based on the simple and computationally efficient 1-D Discrete Wavelet Transform (DWT) via lifting scheme and the Differential Pulse Code Modulation (DPCM). The aforementioned data reduction techniques, in addition to many others proposed in the literature, perform well on stationary univariate time series. However, an important number of IoT devices nowadays include more than one sensor and are able to collect multiple features. Therefore, data reduction techniques that work efficiently on multivariate time series are required.

Compressive Sensing (CS) and transform domain compression techniques that are usually used for images have been proposed as well for multivariate time series compression in IoT applications. In [117], the authors proposed a multisignal compression technique based on the theory of fuzzy transform. The proposed method has been applied on multisignal environmental data collected by a wireless sensor network and reduced the data by approximately two times. In [141], the authors proposed the 2-D lifing wavelet transformation to compress multisignal data collected from different sensor nodes. The proposed method uses the Haar wavelet and achieves a compression ratio of 1.33 and recovery accuracy of 98.4%. Transform domain compression techniques are characterized by the ability to recover the data accurately. However, the compressing performance of these techniques remains limited. On the other hand, CS theory has emerged as an efficient approach for energy-efficiency in IoT applications in recent years [16, 93]. By taking advantage of the signal sparsity, CS approaches assure an accurate signal recovery by sampling signals at a much lower rate than the traditional Shannon-Nyquist theorem. Nevertheless, CS techniques suffer when dealing with non-sparse multi-dimensional signals containing diverse features with different scales of values.

The major drawbacks of the above-mentioned propositions is that they yield low compression ratio on non-stationary multi-sensor data and they are not tested on real devices. This work proposes a data compression technique for IoT applications and resource con-

strained devices that works efficiently on multivariate time series and implemented on real devices.

5.3/ THE PROPOSED DATA COMPRESSION WITH SZ

This section provides a background on the SZ compression method and details its implementation on a Polar M600 smartwatch and low-power system on a chip microcontrollers.

5.3.1/ BACKGROUND

In this work, a lighweight version of the work presented in [142] is given. The authors in [142] proposed a fast error-bounded lossy compression scheme namely $SZqueeze$ (SZ) for High Performance Computing (HPC) applications. This compression scheme has been proposed to deal with the huge amounts of data generated during the execution of HPC applications. SZ is a flexible compressor as compression errors are bounded by absolute or relative error bound and can be managed by the user. Additionaly, SZ can be applied on 1D floating-point arrays and on multi-dimensional arrays. The original SZ compresses input data files that are in binary format and can have different data shapes and data types (single-precision and double-precision). In this work, we propose to adapt the SZ algorithm for IoT devices by considering only the floating point data type and discarding the other types which make the code smaller in size and easier to compile on tiny devices. Moreover, the algorithm was adapted to take a 1-D array of float sensor data as input and return a byte array that is going to be transmitted to the edge node. The motivations behind choosing SZ for IoT applications are as follows:

- SZ allows the compression of multivariate medical time series containing diverse features with different scales
- SZ allows the control of information loss by using an error bound
- SZ leads to higher compression ratio than the multi-dimensional transform domain compression techniques

The proposed compression scheme is defined in Algorithm 1. It is considered that the data are transmitted to the edge after each period P of time t. The collected data are in the form of $M \times N$ array, where M denotes the number of readings and N denotes the number of features. For example, consider a motion sensor that collected 128 gyroscope and accelerometer readings for the three coordinate axes after a period P. In that case, M is equal to 128, and N to 6.

To begin, the 2-D array is converted to the 1-D array (Algorithm 1, line 4). The transformation of a multi-dimensional array to 1D array is the first step of the compression process. The flattened array is then compressed using the lossy SZ technique. Finally, the resulted binary array is transmitted to the edge. Algorithm 2 presents the main steps employed by the adapted SZ compression scheme. Note that the adaptation has been done by extracting the necessary functionalities from the original SZ to make it fit on wearables and resource constrained devices.

Algorithm 1 Proposed compression scheme

Require: E (error bound)
1: **while** $Energy > 0$ AND $Sensors\_status = ON$ **do**
2: **for** each period **do**
3: $data[M, N] \leftarrow$ collected sensors data
4: $input[M \times N] \leftarrow Flatten(data)$
5: $bin\_output \leftarrow adapted\_SZ(input, E, M, N)$ (Alg 2)
6: $transmit\_Data(bin\_output)$
7: **end for**
8: **end while**

Algorithm 2 Adapted SZ steps

Require: input (1-D array), E (error bound),M (num rows),N (num columns)
Ensure: $output$ (binary array)
1: Bestfit Curve-Fitting Compression
2: Compressing Unpredictable Data

The second step of the compression process consists of compressing the 1-D array using adaptive curve-fitting models. The bestfit step employs three prediction models: Preceding Neighbor Fitting (PNF), Linear-Curve Fitting (LCF), and Quadratic-Curve Fitting (QCF). PNF attempts to predict each data point by using the previous data point value. LCF predicts each data point by fitting a line with the previous two consecutive data points, and QCF by fitting a curve with the previous three consecutive values. The adopted model is the one that yields the closest approximation. The estimated data point is replaced by a two-bit code specifying the type of the prediction model. In the case when none of the prediction models in the curve-fitting step satisfies the error bound, the data point is marked as unpredictable and is then encoded by analyzing the IEEE 754 binary representation. (Algorithm 2, line 2). Finally, lossless compression is conducted using algorithms such as Huffman encoding and LZ77.

As for the error bound, the absolute error bound has been used in which the compression/decompression errors are limited to be within an absolute error. For instance, if the value of a data point is considered to be X, an absolute error bound of 10^{-1} means that the decompressed value should be in the range $[X - 10^{-1}, X + 10^{-1}]$.

5.3.2/ LIGHT SZ FOR IOT

The respect for the given error bound and the ability to compress multi dimensional arrays of floating points make this algorithm an excellent candidate for IoT applications. This method, however, runs efficiently on a PC/server in its current form, but not on a highly resource-constrained device. First, SZ was implemented on a wearable device (Polar M600 smartwatch) using Android NDK toolset by slightly adapting the algorithm written in C language. However, things become more complicated when it comes to microcontrollers that are more resource-restricted in terms of memory, space, and computing than wearable devices, more changes should be made to SZ in order to work well on any low-power IoT device.

As explained earlier, SZ is a lossy compression tool dedicated to HPC environment. Several modifications are required to adapt the code on ESP-32 like devices. First, the code is long (around 20,000 lines) and has been designed for 64-bit processors. The first adaptation, therefore, consisted of changing some of the pointers and some types to explicitly force the compiler to use 32-bit pointers. Then, because the code is made for different types of data (double, float, int), only the float case was taken into account. One must bear in mind that the storage is very small on IoT devices, which is why all unnecessary parts have been removed. Many large buffers have been used inside the code, so they have been reduced as much as possible to run the code on ESP-32 chips. Since SZ can use Gzip or Zlib to compress in a lossless manner the output of lossy compression, only Zlib was considered because it is easier to use on Arduino devices. Finally, as it is more difficult to develop codes with arduino than to develop codes on regular machines, it has taken us a lot of time to be able to adapt this code. In fact, the Arduino environment does not have an efficient debugger.

5.3.3/ SZ COMPRESSION FOR LORA SENSOR NETWORKS: MOTIVATION

LoRa (Long Range) is wireless communication protocol which combines ultra-low power consumption with an effective long range. While the latter highly depends on the environment and possible obstructions in the transmission line of sight, LoRa, in general, has a range of tens of Kms, which means a single LoRa gateway can provide coverage for an entire city, and with a couple more, a whole country. The wide range, low power consumption characteristics gives this technology an advantage over conventional power hungry and/or short range communication protocols such as: Zigbee, BLE, WiFi, 3G and LTE.

LoRaWAN is a high capacity, Long Range, open, Low Power Wide Area Network (LPWAN) standard designed for LoRa Powered IoT Solutions. It is a bi-directional protocol which takes full advantage of all the features of the LoRa technology to deliver services

including reliable message delivery, end-to-end security, location and multicast capabilities. LoRa uses a combination of variable bandwidth and spreading factors (SF7-SF12) to adapt the data rate in a trade-off with the range of the transmission. Higher spreading factor allows longer range at the expense of lower data rate, and vice versa. The combination of bandwidth and spreading factor can be chosen according to the link conditions and the level of data to be transmitted. Thus, a higher spreading factor improves transmission performance and sensitivity for a given bandwidth, but it also increases transmission time as a result of lower data rates.

LoRa end-devices have a maximum data rate of around 50kb/s. Compared with most of the other technology this rate is the lowest, which makes it not ideal for certain applications where high data rates are required. Moreover, LoRa-IoT devices are constrained to tight transmission duty cycles. In order to minimize the impact of these limitation on LoRa-based IoT application, data compression can be introduced.

Assuming an ECG sensor operating on a 1000Hz frequency and acquiring 1000 samples per second accounting for 10kb (1250 bytes) worth of data. Assuming also that this sensor is using SF7, and the payload size is 230 bytes (calculate waiting time between each transmission = 36 seconds). Therefore, six packets transmitted over 3.6 minutes are required to transfer the 1250 Bytes to the gateway. Assuming that we compressed the data from 10kb to 1kb (125 bytes). Thus, to transmit the data, only one packet is required, and with approximately 250 bits/s data rate, the transmission time will become 4 seconds.

By compressing the amount of data to transmit we will achieve the following gains: (1) With long waiting period between each transmission forced by the Duty Cycle, the amount of time required to transmit the collected data-set to the gateway is heavily reduced. (2) Since transmission consumes battery, fewer transmissions means lower battery consumption as long as compressing the data-set consumes less than transmitting it.

Therefore, the proposed approach for data compression can be considered as an effective solution for energy-efficiency in LoRa-based applications. The effect of compression on transmission time and energy consumption will be detailed in the experimental results section.

5.4/ CASE STUDY

The combination of IoT, cloud computing, and healthcare has taken a lot of attention during the past years. Among the major challenges that face the healthcare applications are:

- Latency due to communication between cloud and IoT devices.
- Limited network bandwidth due to the high amount of generated data.
- High cost of privacy and security breaches.

Here is where the benefits of edge computing take place. As mentioned in previous sections, edge computing can be considered a major solution for latency and bandwidth challenges in IoT and healthcare applications. In a similar way, efficient on-node data compression can increase the lifetime of the application, and reduce the size and the number of transmitted packets which may affect the latency and network bandwidth positively. However, the integrity of the data and loss of information due to compression remain critical issues for healthcare applications. In this section, the impact of data compression and energy efficiency on the analysis of medical data at the edge is studied. Let us consider the case of driving behaviour monitoring and stress detection where the physiological signals are continuously collected from a driver and transmitted to the edge node for analysis and detection of stress level.

In the following, the dataset used in our work is described, then the processing and analysis of the data are discussed, and finally, the model used for stress level detection is presented.

5.4.1/ DATASET

The Stress Recognition in Automobile Drivers database published on PhysioNet has been used in this work [9, 20]. This multivariate dataset contains multiple physiological signals recorded from healthy volunteers, taken while they were driving on a specified route in and around Boston, Massachusetts. The driving task done by each driver varies from about 50 min to 1.5 h and can be divided into six sections as shown in Figure 5.2:

- Rest [1,6]: The resting periods in the beginning of the driving task and at the end of it are labeled as ”low stress”.
- City [2,5]: Driving in the city periods in sections 2 and 5 are labeled as ”high stress” since the subjects drove in a busy main street and frequently handled the traffic conditions and the unexpected emergencies created by cyclists and jaywalkers.
- Highway [3,4]: Driving on the highway periods in sections 3 and 4 are labeled as ”moderate stress”.

Note that the labeling of the data has been validated by drivers’ self-reported questionnaires in [20].

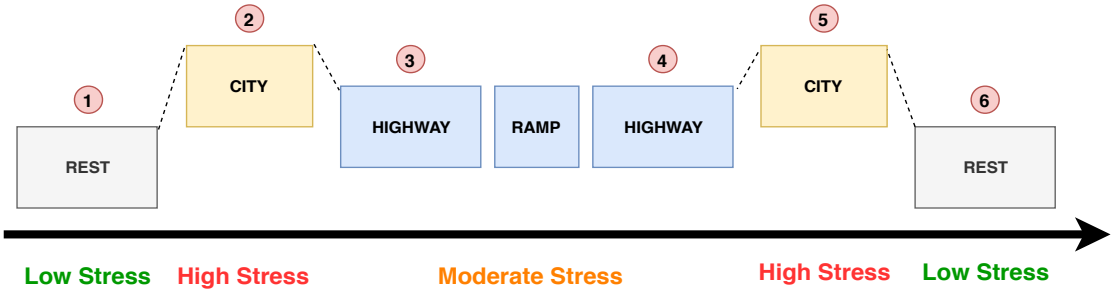


Figure 5.2: Driving task.

This study is conducted on 9 drivers among 17 available in the database since the marker indicating the driving sections (rest, city, highway) is not present in all drivers files. Five physiological signals were used for stress detection, namely electrocardiogram (ECG), heart rate (HR), galvanic skin response (GSR) of the hand and foot, and respiration rate (RR).

5.4.2/ Data processing and analysis

For each driver data file, the noise has been removed from HR and GSR signals following [124]. In order to study the impact of data compression on the information contained in the data, a feature extraction phase is used from which time-domain and frequency-domain features are extracted from the ECG signal in addition to the two main components of the GSR signal namely: Skin Conductance Level (SCL), and Skin Conductance Response (SCR). The objective of this step is to compare the features extracted from the original data with the features extracted from the compressed data.

Table 5.1 describes the features extracted from the ECG signal by applying time and frequency domain analysis such as FIR-filters and fast fourier transform methods. In order to analyze the GSR signal, which can be considered an important sensitive measure for emotional arousal, we extract the slow variation (SCL) and the faster alterations (SCR) following a convex optimization approach proposed in [148]. Table 5.2 shows the features extracted from the GSR signal.

5.4.3/ Stress detection using Feed-Forward Neural Network

This section considers the features shown in Tables 5.1 and 5.2 in addition to heart rate (HR) and respiration rate (RR) as a single sequence in which a label is assigned. The prediction task is a supervised sequence classification task. A FFNN is a network that contains a large number of neurons, arranged in layers: one input layer, one or more hidden layers, and one output layer.

Feature	Description
RMSSD	Root mean square of the Inter-beat (RR) Intervals (the time intervals between consecutive heart beats)
meanNN	Mean RR interval
sdNN	Standard deviation RR interval
cvNN	Coefficient of Variation (CV), i.e. the ratio of sdNN divided by meanNN
CVSD	Coefficient of variation of successive differences, i.e. the RMSSD divided by meanNN
medianNN	Median of the absolute values of the successive differences between the RR intervals
madNN	Median Absolute Deviation (MAD) of the RR intervals
mcvNN	Median-based Coefficient of Variation (MCV), i.e. the ratio of madNN divided by medianNN
pNN20	The number of interval differences of successive RR intervals greater than 20 ms divided by the total number of RR intervals
pNN50	The number of interval differences of successive RR intervals greater than 50 ms divided by the total number of RR intervals

Table 5.1: Features extracted from ECG signal.

Feature	Description
meanGSR	Mean value of GSR signal
meanSCL	Mean value of SCL
slopeSCL	Difference between max and min values of SCL
meanSCR	Mean value of SCR
maxSCR	Max value of SCR

Table 5.2: Features extracted from GSR signal.

Figure 5.3 shows the FFNN architecture used in the classification task. The neural network consists of 17 input neurons, corresponding to a sequence of 17 physiological features. Additionally, the neural network is provided with a correct label of that sequence. That is, whether the sequence corresponds to low stress, moderate stress, or high stress. Providing that edge analytics require lightweight algorithms to perform reasonable machine learning [163], we have used a network with a minimized size that maintains accuracy. The implemented network has four hidden layers consisting of 60 neurons, which use the ReLU activation function [49]. The number of layers The network was trained with a variant of stochastic gradient descent 'Adam' [96] and categorical cross entropy. Furthermore, Dropout regularization technique [106] has been used to prevent overfitting and 10-Fold Cross Validation has been used to validate and find the optimal set of hyperparameters for the model. Finally, the network has been trained for 300 epochs with a

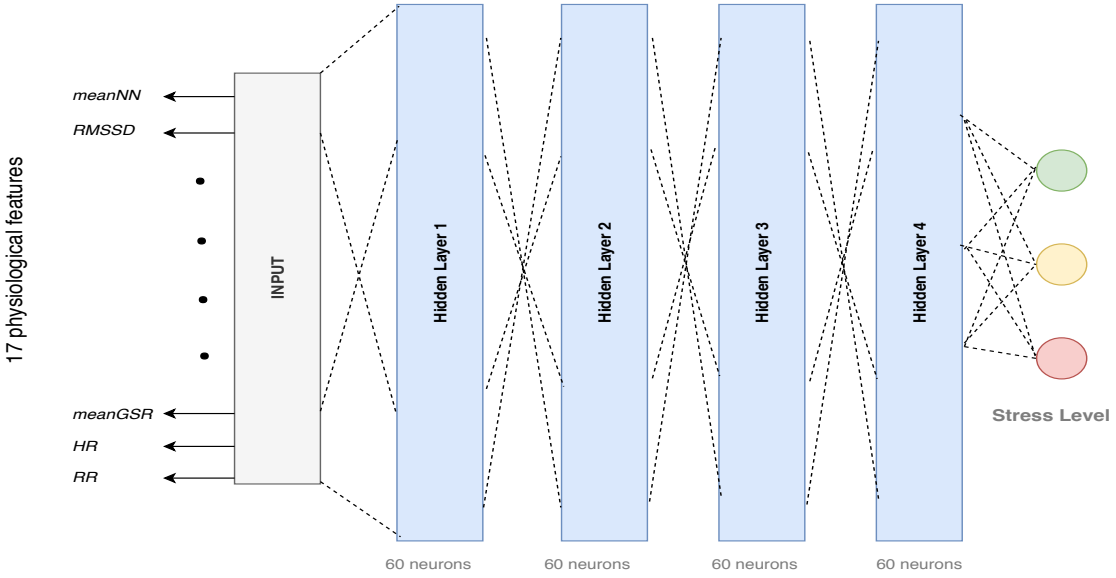


Figure 5.3: Feed Forward Neural Network.

learning rate of 0.01[1].

5.5/ EXPERIMENTAL SETUP

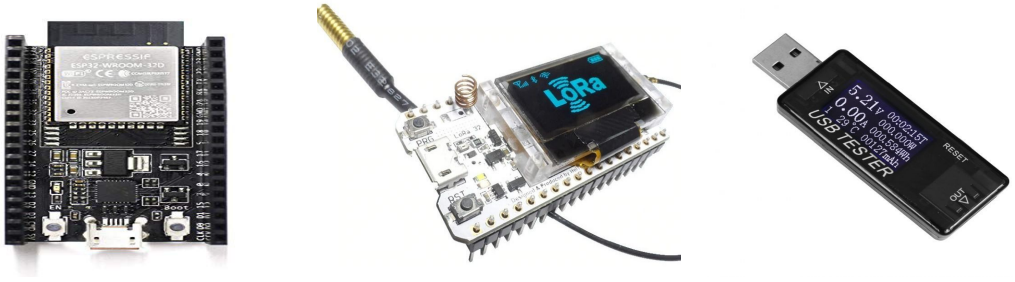
This section describes the tools and datasets used to evaluate and validate the proposed approach by measuring the effect of data compression on the IoT devices' energy consumption and the time required for processing and transmitting the collected data. To do so, the ESP Wroom 32 and Wifi LoRa 32 were used to test the transmission with WiFi and LoRa.

ESP Wroom 32 is a WiFi+Bluetooth microcontroller unit module capable of targeting applications for low-power sensor networks. WiFi LoRa 32 is a popular IoT dev-board manufactured by Heltec Automation (TM), a highly integrated system based on ESP32, featuring WiFi, bluetooth, and LoRa capabilities and suitable for smart city applications. In order to assess the impact of processing and transmission on energy, a USB Tester has been used to measure USB voltage and current drawn by USB devices. USB tester helps to measure the amount of energy consumed in mWH by a device plugged into a USB port. These devices are shown in Figure 5.4.

In addition to the multivariate Stress Recognition in Automobile Drivers dataset discussed in the previous section, two other univariate datasets were also considered. The first is the ECG-ID Database that was published in Physionet [9] and originally proposed in [241]. The second is a PPG dataset that was recorded in the Femto-ST laboratory department

[1]The following example illustrates how features can be extracted from raw physiological signals and used as input to an artificial neural network: https://josephazar.net/blog/mlmitdriver.php

DISC, Belfort, France, using Shimmer3 GSR+Unit [237].



(a) esp wroom 32 (b) Wifi LoRa 32 (c) USB tester

Figure 5.4: Captures of the materials used for measuring the effect of data compression on energy consumption and processing/transmission time.

In addition to the aforementioned devices, a Polar M600 smartwatch has been used as well to test the impact of data compression on its battery lifetime.

5.6/ Experimental results and analysis

This section describes the results of the experiments conducted to validate the solution proposed in this chapter. Four metrics are discussed in the following: data reduction, processing and transmission time, energy consumption, and the impact of information loss on the prediction accuracy.

5.6.1/ Data reduction

This section shows the results obtained after compressing different batches of ECG and PPG data using SZ with three different error boundaries: 10^{-1}, 10^{-2}, and 10^{-3}. The ECG and PPG batches used in this section come from the ECG-ID dataset and the locally-collected PPG dataset respectively. Note that using SZ with an error bound e means that the decompressed data points d are within the range $[d-e, d+e]$.

Most of the sensors deployed in real world applications capture and transmit data periodically. The experiment conducted simulates this approach and consists of compressing a 10Kb batch of data after each period of time $t = 30s$ and logging the compressed size for 80 periods. This process was repeated using the three error-bounds for ECG and PPG data compression. The results are shown in Figure 5.5.

The results clearly show that increasing the error bound results in a higher compression ratio. The ECG and PPG data compression ratio obtained by using SZ with 10^{-1} error bound ranges from 80:1 to 100:1. When using an error bound of 10^{-2}, the compression

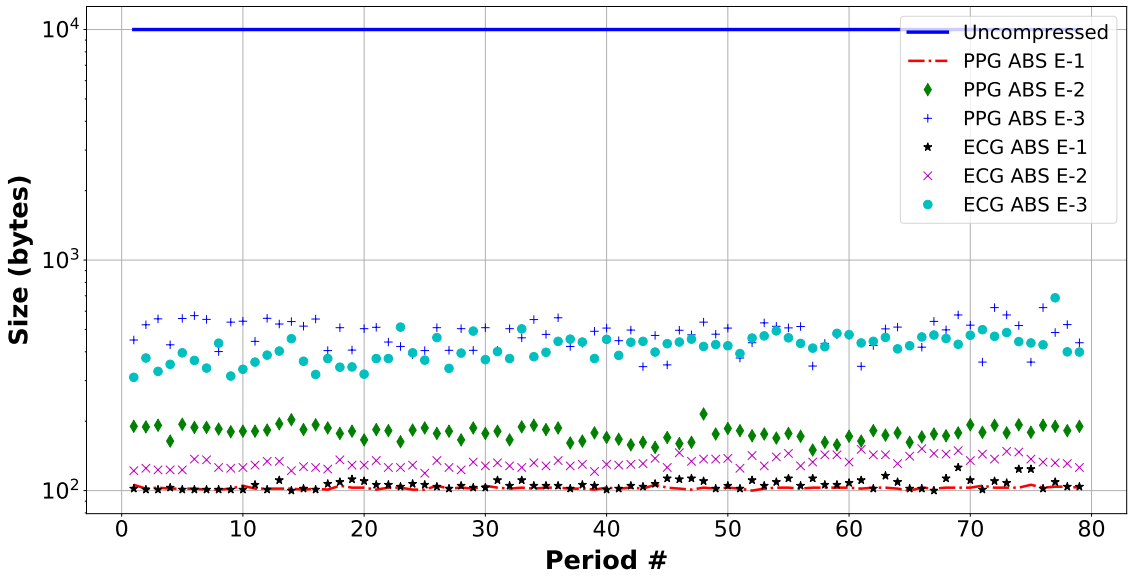


Figure 5.5: Compression of 10000 bytes of ECG and PPG data using SZ with different absolute error bound values over 80 periods.

	SZ E-1	SZ E-2	SZ E-3
PPG	97:1 ± 3	56:1 ± 10	20:1 ± 6
ECG	94:1 ± 13	75:1 ± 9	23:1 ± 9

Table 5.3: Compression ratios achieved by SZ using three different error-bounds on ECG and PPG datasets.

ratio ranges from 45:1 to 85:1 and falls to an average of 20:1 when using an error bound of 10^{-3}. These values are shown in Table 5.3.

Although the objective of the work in this chapter is to increase the lifetime of IoT devices, the remainder of our experiments use SZ for data compression with an absolute error bound of 10^{-1}.

5.6.2/ Processing/Transmission time

A major advantage of data compression is to reduce the time it takes to send the data to the edge/gateway, which is a critical requirement in near-real-time applications. This section compares the time required for transmitting uncompressed data and the time required for compressing and transmitting compressed data. To do so, the Mackay-Glass time series was generated on the ESP Wroom 32 and the time needed to transmit various sizes of this time series using WiFi to a local PC was calculated. The same procedure was conducted for the time series compression and transmission and it was repeated 20 times for each data size. Figure 5.6 shows the average time required in milliseconds to transmit the original data and to compress and transmit the compressed data to a

gateway.

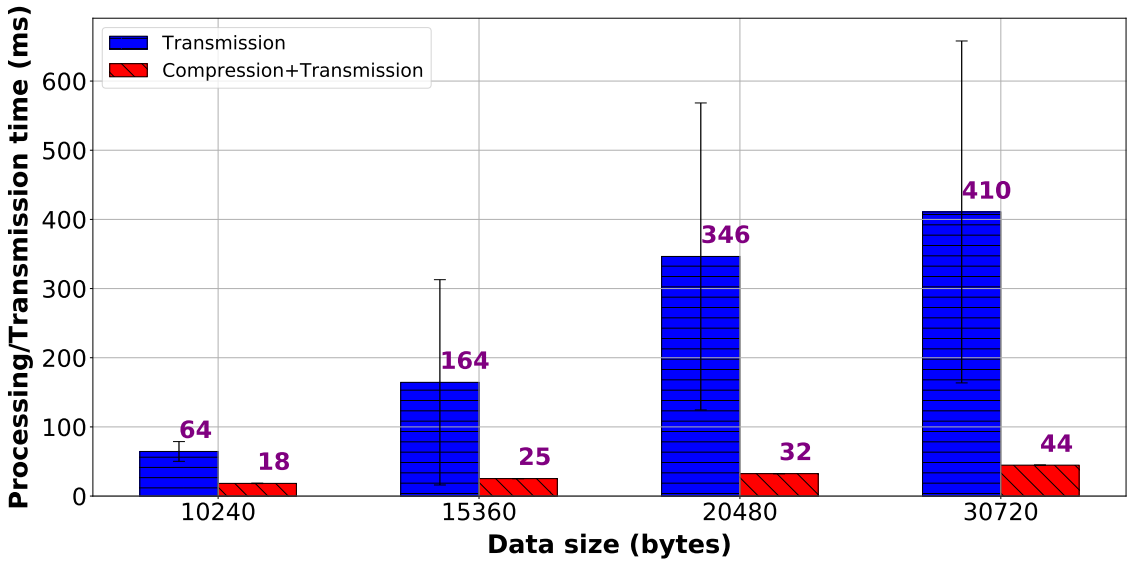


Figure 5.6: Comparing the time needed to transmit data and the time needed to compress and transmit data in ms.

The results show that using the proposed compression technique prior to transmission provides a time gain of 3.5 to 10 times as compared to directly transmitting the original data. Additionally, the results show that data compression becomes more effective with the increase in data size. This will allow for a larger time period t for periodic applications, thereby collecting and sending more data with fewer transmissions.

5.6.3/ Energy consumption

The effect of data compression on energy consumption is discussed in this section. The proposed compression technique was implemented on an ESP Wroom 32, a WiFi LoRa 32, and a Polar M600 smartwatch. The USB tester was used to estimate the energy consumption of the microcontrollers. Three communication protocols have been considered in this experiment: WiFi, LoRa, and Bluetooth Low Energy (BLE).

5.6.3.1/ WiFi

The first experiment is to send different sizes of the generated Mackay-Glass time series to a local PC using WiFi from the ESP Wroom 32. Three cases are considered: (1) transmission of uncompressed data, (2) compression of data with SZ, (3) transmission of compressed data. Each case was repeated 2500 times in a row for each data size and at the end the energy consumption in mWH was recorded. Note that between consecutive transmissions, the sensor does not enter a sleep phase.

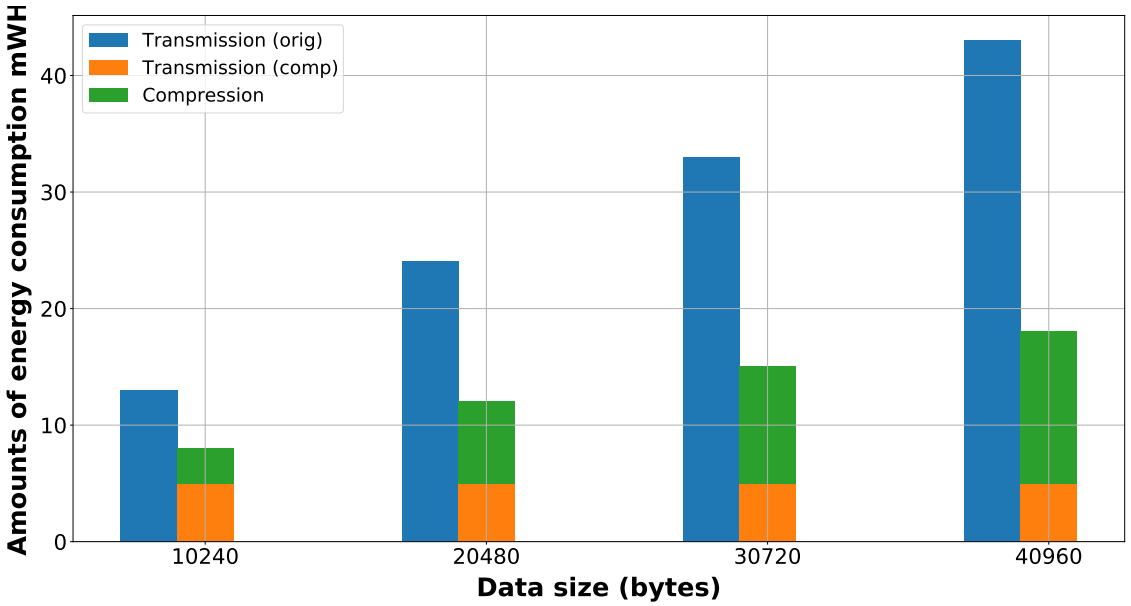


Figure 5.7: Amounts of energy consumption in mWH for original data transmission, compressed data transmission, and compression after 2500 cycles.

Figure 5.7 shows that the energy required to perform 2500 transmissions of 10240, 20480, 30720, and 40960 bytes of floating points data is around 13, 24, 33 and 43 mWH respectively. On the other hand, the energy required to compress and transmit 10240, 20480, 30720, and 40960 bytes of floating points data 2500 times is around 8, 12, 15, and 18 mWH respectively. It can be noted that the energy needed to transmit the compressed data remains stable (5 mWH) with the increase in data size while the processing energy increases slightly with the size of the data.

The ratio of energy savings varies from about 1.5 to 2.5. For example, using the proposed approach to transmit 40Kb can reduce energy consumption by up to 2.5 times compared to uncompressed data transmission. It is possible to deduce the same conclusion as the previous section, i.e. increasing the period time t and collecting more information, then compressing and transmitting, is the optimized method which allows for the highest energy and storage gain.

5.6.3.2/ LoRaWan

The second experiment is to send different sizes of the generated Mackay-Glasstime series to a LoRaWan Gateway. Similarly, to the first test we consider the following cases:

- The raw data are transmitted without any compression
- Data are compressed before transmission.

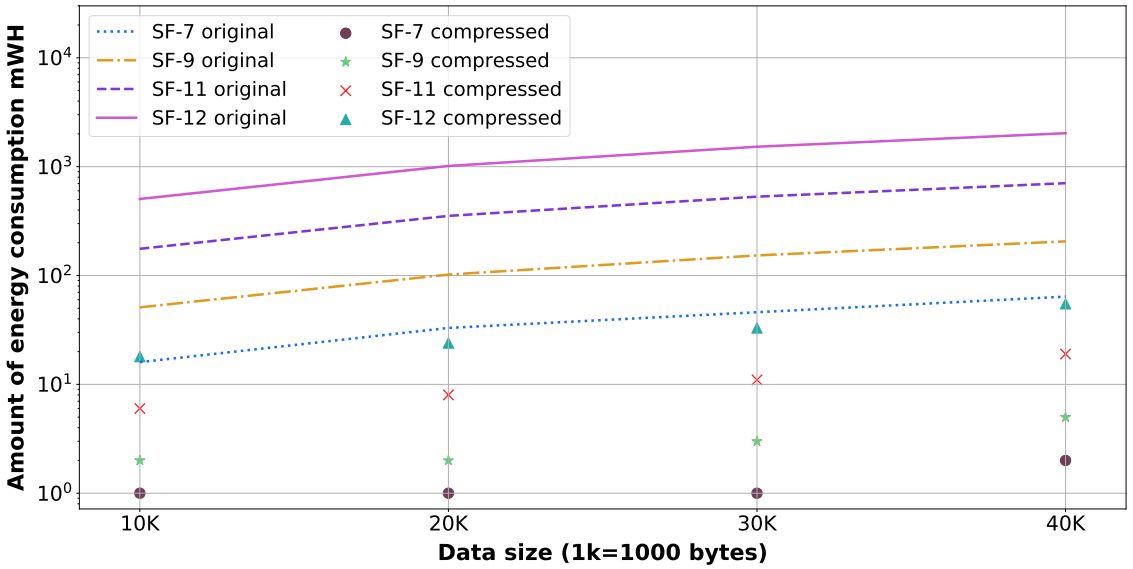


Figure 5.8: The effect of SF and data size on the energy consumption.

Each case was repeated 10 times, and the final results seen in Figure 5.8 are the averaged values. Unlike Wifi and in order to respect the Duty Cycle of the LoRaWan, the device is forced to transmit a limited amount of data periodically after a certain amount of time is passed. Meanwhile, while the device is not allowed to transmit, it is put to sleep.

The allowed size of the transmitted packet, in addition to the sleep duration depends on the used Spreading Factor (SF). Detailed information can be seen in Table 5.4.

Data rate	Configuration	bits/s	Max payload size (Bytes)	Time on Air (ms)	Duty Cycle (mm:ss)
DR0	SF12/125kHz	250	59	2629.63	04:23
DR1	SF11/125kHz	440	59	1478.66	02:28
DR2	SF10/125kHz	980	59	657.41	01:06
DR3	SF9/125kHz	1760	123	656.38	01:06
DR4	SF8/125kHz	3125	230	635.39	01:04
DR5	SF7/125kHz	5470	230	363.78	00:36

Table 5.4: LoRa experimental settings.

The energy consumption varies greatly depending on the SF. The higher the latter is, the lower the data rate is, which forces the radio module to operate for a longer duration in order to transmit the packet and increases the transmission energy. In addition, the packet size gets smaller for higher spreading factors. As a consequence, more packets are needed to transmit the same total amount of data, therefore, more energy is consumed in the process. Finally, while the sensor is in sleep mode it still consumes some energy. The higher the spreading factor is, the higher the time on the air (ToA) of the transmitted packet is, which means the sensor is forced to wait longer for the next transmission. The

more time the sensor waits to transmit the stored data, the more it consumes.

All of this can be seen in Figure 5.8, the higher the SF is the more energy is consumed. But most importantly, we can see the impact of compression on the energy consumption. Indeed, compressing the data before transmission reduces drastically the amount of consumed energy.

5.6.3.3/ BLE

The multivariate Stress Recognition in Automobile Drivers dataset has been used to study the energy consumption on the wearable smartwatch. The signals in this dataset were recorded from the drivers at 496 Hz. In order to test the efficiency of the proposed compression scheme, we have deployed the data on the SD card of a Polar M600 wearable. In the same way, the compression technique written in C language using Android NDK toolset was implemented. After each period $p = 1\,min$, an array $data[M, N]$ is transmitted using Android BLE to the edge device, which is a local PC where the received data are processed and analyzed. The results are then transferred to the cloud. Note that M, the number of readings, is equal to 148800 and N, the number of features, is equal to 5. The experimentation has been conducted for around 4 hours, equivalent to 241 periods.

Figure 5.9 describes the change of the wearable battery level over 241 periods for five different scenarios:

- Black line: the wearable device is in the idle state.
- Red line: the wearable device is continuously collecting data (motion and heart rate sensors are turned on).
- Yellow line: the wearable device is collecting data and running the compression algorithm after each period.
- Green line: the wearable device is collecting data, and performing compression and transmission after each period.
- Blue line: the wearable device is collecting data, and performing data transmission after each period (no compression).

The results clearly show the impact of data reduction on the communication task energy consumption. The battery level of the device continuously collecting data was decreased to around 86% after 241 periods. Note that by comparing the sensing and computation tasks with the idle state, it can be noticed that these tasks contribute a little in the energy consumption of the device. On the other hand, when applying the proposed data compression scheme prior to transmission, the battery level was decreased to 83% while

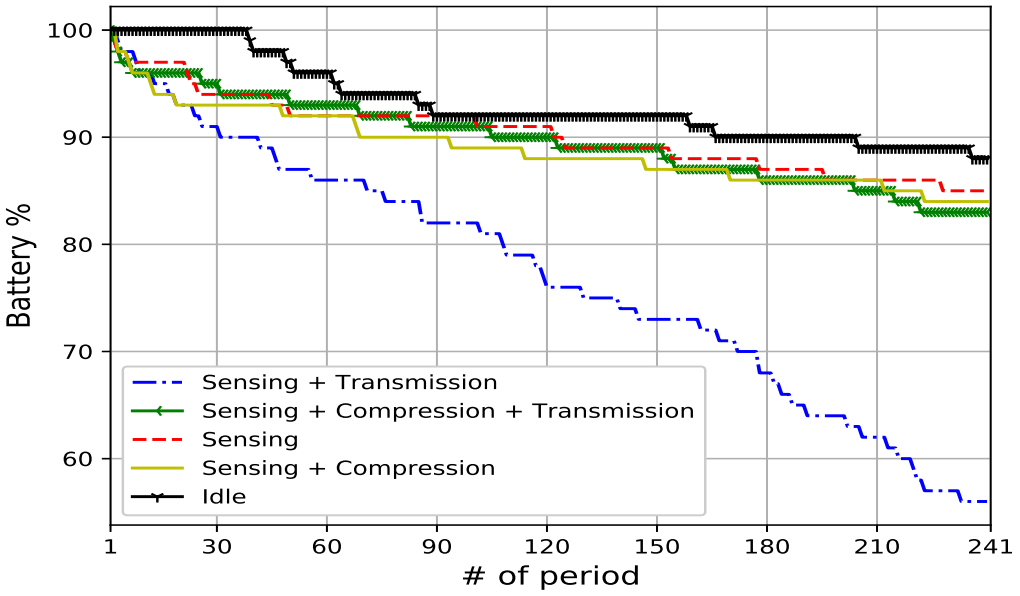


Figure 5.9: Polar M600 battery level over 241 periods.

sending the data without compression decreased the battery level to around 56%. As a result, the device lifetime could be increased by up to 27% after 4 hours.

5.6.4/ Loss of information and stress detection

In this section, the impact of information loss on the prediction accuracy of the FFNN model for stress level detection is studied. For each of the drivers datasets, a sliding window of 30 seconds with 75% overlap on the data is applied, and the physiological features described in the previous section are extracted from each window. Note that for each window, the extracted features form a set/sequence labeled as low, moderate, or high stress that is going to be fed as an input to the neural network.

Figure 5.10 compares the original ECG signal transmitted to the edge without compression with the compressed signal using an error bound of 10^{-1}. Even though the compression has affected the shape of the signal, the positions of the peaks that are used to exploit the signal and extract the most important features remained unchanged. Note that one of the advantages of the SZ algorithm is that the error bound is controllable and can be initialized according to the medical need. In other words, the trade off between compression ratio and loss of information can be easily controlled by the medical staff.

Table 5.5 shows the Root Mean Square Error (RMSE) between the features extracted from the compressed and original ECG and GSR signals for the 9 drivers. Notice that the average RMSE for each feature is small, and the important features such as heart

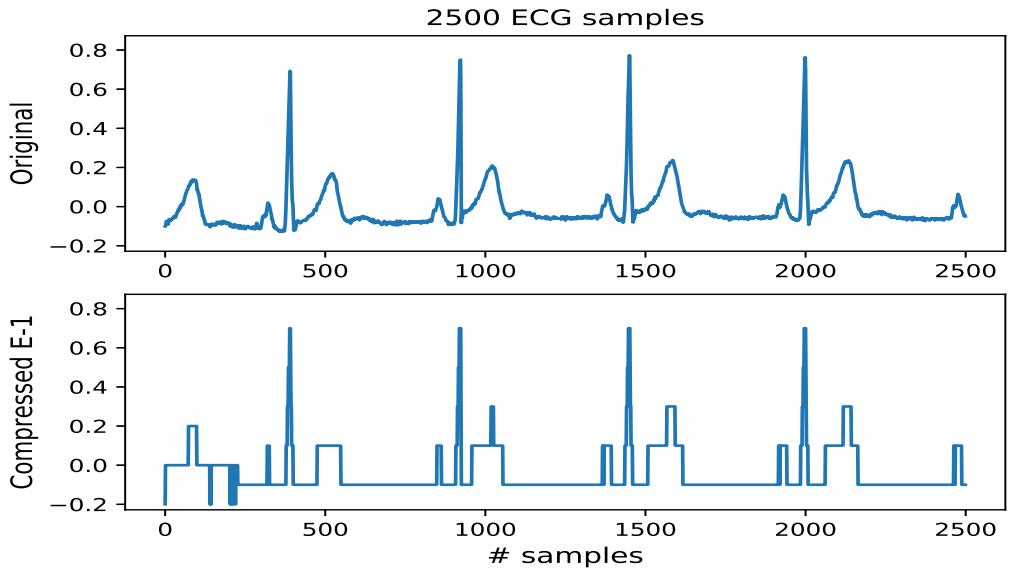


Figure 5.10: 2500 original ECG samples vs compressed ECG samples. Dataset: Stress Recognition in Automobile Drivers [20]

	Driver #									*Average*
	4	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**16**	
HR	0.27	0.08	0.07	0.10	0.04	0.08	0.07	0.07	0.26	**0.11**
RMSSD	1.57	1.68	1.18	0.94	1.20	2.45	3.77	2.28	2.42	**1.94**
meanNN	0.91	1.24	0.24	0.23	0.25	1.02	0.30	0.70	1.09	**0.66**
sdNN	0.61	1.01	0.29	0.33	0.58	1.12	1.65	0.87	1.44	**0.87**
cvNN	0	0	0	0	0	0	0	0	0	**0**
CVSD	0	0	0	0	0	0	0	0	0	**0**
medianNN	2.19	1.4	0.67	1.34	1.01	2.08	1.74	1.06	1.63	**1.45**
madNN	1.91	1.19	1.01	1.22	1.12	1.96	1.95	0.88	1.52	**1.41**
mcvNN	0	0	0	0	0	0	0	0	0	**0**
pNN50	1.78	2.2	1.23	1.51	1.91	1.92	2.24	1.56	1.27	**1.72**
pNN20	2.52	3.63	1.70	1.95	1.93	2.39	1.80	1.92	3.38	**2.35**
mean_gsr	0	0	0.01	0.02	0.02	0.01	0.02	0	0	**0**
mean_scl	0.23	0.22	0.23	0.21	0.21	0.20	0.20	0.19	0.23	**0.21**
slope_scl	0.38	0.36	0.34	0.34	0.28	0.33	0.34	0.29	0.44	**0.34**
mean_scr	0.22	0.22	0.23	0.21	0.21	0.20	0.20	0.19	0.23	**0.21**
max_scr	0.37	0.31	0.30	0.31	0.23	0.30	0.23	0.27	0.41	**0.30**
RR	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	**0.02**

Table 5.5: Root mean square error between the features extracted from original data and the features extracted from reconstructed data. Dataset: Stress Recognition in Automobile Drivers [20]

rate, respiration rate, and R-R interval have a RMSE close to zero, which means that the compression had very low impact on the information loss.

In order to fully answer the stated problem in the first section, the sequences of features

	Accuracy	
	Original	***Compressed***
Driver 4	1.0	0.99
Driver 6	1.0	1.0
Driver 7	0.97	0.98
Driver 8	0.98	0.98
Driver 9	0.98	0.97
Driver 10	0.99	0.99
Driver 11	0.92	0.94
Driver 12	0.99	0.99
Driver 16	0.99	0.98
Average	**0.98**	**0.98**

Table 5.6: FFNN prediction accuracy on test sets (25%) corresponding to sequences of features extracted from original and compressed data respectively. Dataset: Stress Recognition in Automobile Drivers [20]

are randomly divided into train and test sets (75%/25%). Two models are considered, the first one was trained on the sequences of features extracted from the original data and the second one on the sequences of features extracted from the compressed data. For hyperparameter optimization, 10-fold cross validation is performed on the training sets, and then our models are evaluated on the test sets. Table 5.6 summarizes the stress level detection performance of the aforementioned models. The results show that the average accuracy achieved by the two models is 98%. It can be noticed that not only the compression did not affect the prediction accuracy but even improved it in some cases such as for driver 7 and driver 12, which is due to the denoising capability of the compression technique.

5.7/ LIMITATIONS OF THE PROPOSED WORK

The results of this work have shown that the proposed compression technique has been able to achieve satisfactory results in terms of energy conservation and data reduction. For maximizing the compression ratio, SZ with an absolute error bound of 10^{-1} was used and Section 5.6.4 showed that the impact of data compression on data quality and the values of the extracted features was not significant. However, after compression with SZ E-1, the features extracted from the Stress Recognition in Automobile Drivers dataset [20] were extracted from the ECG-ID [241] and the locally-collected PPG datasets as well. The results showed that the hypothesis of using SZ with maximum compression ratio in different scenarios can not produce consistent results.

Contrary to the results obtained in Table 5.5 when the drivers stress recognition use case was considered, the average values of the extracted features from the reconstructed

ECG-ID and the locally-collected PPG datasets were not plausible. It can be seen that the the compression with SZ with an absolute error bound of 10^{-1} resulted in certain features such as RMSSD and sdNN being unreliable and entirely different from those extracted from the original datasets.

	RMSE	RMSSD	meanNN	HR	pNN20	pNN50	sdNN
Original PPG		4.33	999.44	60	0	0	2.89
Reconstructed PPG	0.05	64.50	997.48	60	0.76	0.61	41.09
Original ECG		1.81	1000	60	0	0	1.09
Reconstructed ECG	0.05	116.51	974.06	68	77.27	63.63	85.28

Table 5.7: Comparison of the average values of the root mean square error and the features of the original and reconstructed signals. Datasets: ECG-ID [241] and locally-collected PPG.

Another limitation of the reconstructed signal is its graphical representation and form. Even though the peak positions were not affected as shown in Figure 5.10, when compressed with a high compression ratio, a medical doctor or physician may not accept the visual representation of the signals.

The straightforward possible solution to these limitations is considering a slightly smaller error-bound such as 10^{-2} or 10^{-3}. So if the question stated in Section 5.1 is to be addressed, we may conclude that while the use of a lossy compressor in IoT-healthcare applications may help solve several challenges, it does not guarantee accurate data restoration in all scenarios when it comes to data quality. These limitations will be further discussed and addressed in the next two chapters.

5.8/ CONCLUSION

Since bringing intelligence closer to IoT devices reduces network latency and energy consumption due to radio communications, data reduction can be seen as an additional solution to increase the lifetime of IoT devices. In this work, an energy efficient data reduction scheme for IoT-Edge applications in healthcare was proposed. The proposed scheme is based on an error-bounded lossy compressor designed for high performance computing applications that produce large amounts of data during the execution. A lightweight version of the Squeeze (SZ) compressor have been implemented on real IoT hardware and a Polar M600 wearable. The performance of the compressor was tested on one medical multivariate time series dataset related to automobile drivers stress and two physiological univariate time series datasets. Furthermore, we considered the use case of drivers stress recognition and studied the impact of lossy data compression on the analysis, exploit, and classification of medical data.

The results showed that the proposed solution could reduce energy consumption by as

much as 2.5 times. In addition, compressing large batches of data prior to transmission will reduce the time required to transfer data from the IoT to the edge. It has been shown that the proposed solution has provided up to 10 times gain in time compared to the direct transmission of uncompressed data. As a result, what is proposed in this chapter can be considered for a wide range of IoT applications requiring an easy-to-implement algorithm that reduces the processing/transmission time and optimize the lifetime of the devices.

6

IoT Time Series Classification with Data Compression and Deep Learning

Internet of Things (IoT) and wearable systems are very resource limited in terms of power, memory, bandwidth and processor performance. Sensor time series compression can be regarded as a direct way to use memory and bandwidth resources efficiently. On the other hand, the time series classification has recently attracted great attention and has found numerous potential uses in areas such as finance, industry and healthcare. This chapter investigates the effect of lossy compression techniques on the time series classification task using deep neural networks.In Chapter 5 it was shown that the optimal data quality is not always guaranteed when using a compressor with maximum compression ratio, therefore the trade-off between compression ratio and data quality must be further investigated. This chapter proposes an efficient compression approach for univariate and multivariate time series that combines the lifting implementation of the discrete wavelet transform with the error-bound compressor presented in Chapter 5, namely *Squeeze* (SZ), to attain an optimal trade-off between data compression and data quality.

6.1/ Introduction

The world is becoming increasingly connected and intelligent through the Internet of Things (IoT), which can be defined as a wireless network of smart sensing devices connected to the Internet, ready to collect and transmit data supported by embedded devices or sensor networks. IoT devices are generally composed of five main components: 1) sensors, 2) processor, 3) memory, 4) communication module, and 5) battery. Many of these devices are connected by a gateway functionality that is capable of communicating with the sensors and providing storage and processing capabilities. This gateway could

be located in the cloud or at the edge.

Time series data collected from IoT devices such as sensors, machines, and wearables, are becoming the most widespread. These data are generated with high velocity from different real-world applications such as healthcare, manufacturing, agriculture, and urban infrastructure [109], and transmitted to cloud or edge processing.

One of the main issues with IoT applications is the huge volume of collected data. The on-board signal processing and compression algorithms could be used to avoid this problem and therefore avoid the transmission and storage of large amounts of data. Generally, radio communication dominates energy consumption. Therefore, data compression can significantly decrease communication energy costs by reducing the number of bits to be transmitted and thus increase the lifetime of the device [83]. Consequently, this directly affects the energy consumption and battery life of the devices in addition to the storage requirements of the transmitted data, enabling much less data to be managed.

Data compression algorithms can be divided into two types: lossless and lossy compression. Lossless compression does not involve data loss, while lossy compression leads to data loss. However, the maximum level of compression that can be achieved without loss is limited. This limited compression ratio is unfortunately a big drawback in the context of IoT data and resource constrained devices covered in this work. On the other hand, lossy techniques cannot reconstruct the original data from the compressed data. This does not mean, however, that reconstructed data are of low quality. A good lossy compressor maintains the critical information while dropping, for example, the relatively useless data such as noise.

While on-board lossy compression techniques can increase the lifetime of IoT devices and enormously reduce the amount of data sent to the gateway, the impact of compression ratios on data analysis can not be neglected. The time series collected from IoT applications are transferred to the edge or cloud for processing. Time Series Classification (TSC) is a major and challenging problem for mining received data. TSC research have found a large number of potential applications in areas such as finance, healthcare, and industry. Deep Neural Networks (DNNs) are among the promising techniques used recently for TSC [201]. The main advantage of end-to-end DNN models over state-of-the-art approaches is their lack of heavy preprocessing on raw data.

In an IoT Big Data architecture, time series data pass through different phases. An example of IoT Big Data architecture can be composed of four layers as shown in Figure 6.1 [159]. In the first layer, time series data are generated from different IoT sources and then collected and compressed prior to transmission. The second layer provides communication between IoT devices and gateways using different communication technologies. The third layer is the main layer of the entire analytical system that processes the data using tools such as Apache Storm and Spark. In this layer, the classification

of time series data takes place. The fourth layer is responsible for the interpretation and use of the generated results. In real world applications, time series data are highly compressed in layer 1 before their processing in layer 3.

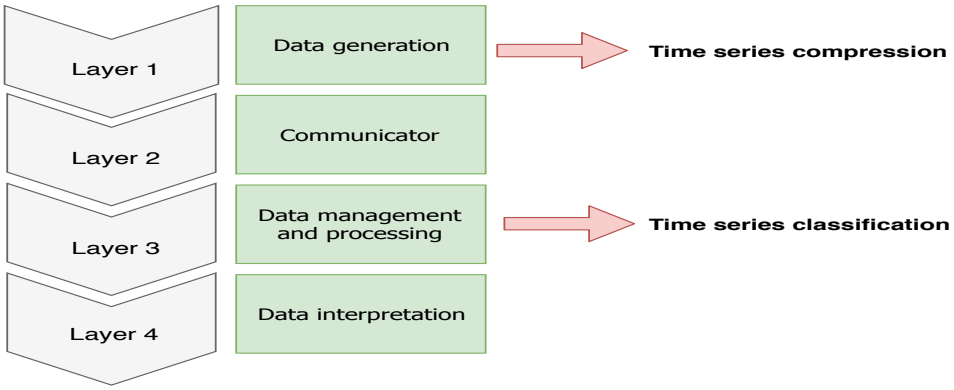


Figure 6.1: Example of IoT Big Data architecture.

Most of the proposed compression techniques for IoT in the state-of-the-art focus on compression ratio, distortion level, and energy consumption in the first layer and ignore classifier performance in the third layer. The work in this chapter focuses on time series compression and classification tasks, and studies the impact of the compression task on classification task in IoT applications. The stated problem in this work can be formulated as follow: *How does the use of time series lossy compressors on IoT nodes affect the classification performance of deep learning models?*

Given the severe resource constraints of IoT nodes, the following metrics must be considered by a compression technique targeting IoT application characteristics: (1) Transmission saved energy must be higher than the energy consumed for processing. (2) Limited on-chip memories are used by embedded processors on IoT nodes. (3) The compression algorithm can be adapted to any application, sensor or activity. (4) Suitable for near real-time applications. (5) Easy to deploy and adapt to various nodes. (6) High compression ratio capability while maintaining data quality. (7) Ability to handle multi-sensor readings on one device.

One of the main contributions in this work is the comparison of three well-used lossy compression techniques in the literature, namely compressed sensing, discrete wavelet transform, and error-bounded compression in terms of compression ratio and classification performance impact.

On the basis of this comparison, this chapter proposes a new approach that combines advantageous properties of the discrete wavelet transform with the presented SZ (SZ) compressor, taking into account all the above-mentioned metrics and requirements.

6.2/ RELATED WORK

Different data reduction techniques are proposed in the literature to resolve energy constraints [70, 83]. In a recent work, Blalock *et al.* [196] presented an efficient high-ratio lossless compression technique for multivariate integer time series, namely Sprintz, with a significantly lower memory and latency than state of the art methods. Compared to different compressors, like SIMD-BP128 [121], FastPFOR [121], Zstandard (Zstd) [170] and Zlib [3], the proposed algorithm achieved strong compression ratio, speed and memory requirements. The authors in [135] proposed a dynamic lossy compression approach to extract valuable information from IoT data with constant adaptation and information loss. The proposed technique is based on the lossy Chebyshev compression and yields a compression ratio of up to 3:1. Various compression algorithms have been proposed that take advantage of the temporal correlation in the data. A new principle for delta compression was developed by the authors in [187], which allows the compression of tempo-related data and results in a higher compression ratio and a lower memory capacity than standard delta coding. Dictionary-based compressors have been proposed to benefit from repetitive patterns in the data. In [208], the authors proposed a lossless data compression technique called Differential-Lempel-Ziv-Welch (D-LZW), which consists in finding the difference between two successive data samples before placing the data in the LZW algorithm. The authors compared their approach to bzip2 and gzip, showing better results in compressing electrocardiogram (ECG) data.

With regard to the problem stated in this work, the influence of data reduction on classification performance was investigated in previous studies. In [218], the authors examined the effect of compressed sensing and reconstruction algorithms like Basis Pursuit and Orthogonal Matching Pursuit on ECG arrhythmia detection using Support Vector Machine (SVM) classifier. The results show that the classification performance remains stable until a compression ratio of 7:1, after which Basis Pursuit outperforms Orthogonal Matching Pursuit on ECG arrhythmia detection. In a similar work [43], the authors proposed the compressed sensing of time series for the recognition of human activities. The results show that the use of compressed sensing on the original time series increased the accuracy of classification using Hidden Markov Model from 77% to 95%. The explanation for this increase in classification accuracy is that the compressed sensing has some kind of advantageous dimensionality reduction, which facilitates pattern recognition. In web page classification, the authors in [17] used the idea of data reduction to make both summarized and pure data available as inputs for the classifier. Their study showed that human or machine summaries help improve the performance of the web page classification.

In the following sections, the influence of three compression techniques, namely Compressed Sensing (CS), Discrete Wavelet Transform (DWT), and Squeeze (SZ) proposed in [142] and presented in Chapter 5, on time series classification is studied. Moreover,

the proposed data compression approach is presented.

6.3/ TSC USING DEEP LEARNING

Because time series data have to be properly classified, researchers have proposed numerous approaches based on deep learning models to solve this task. Different studies of the latest DNN architectures for TSC were conducted in [201, 222, 223, 224], that examine the current state of the art of deep learning algorithms for TSC. Although many kinds of DNNs exist, two principal DNN architectures used for TSC have been used in this work: Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN).

6.3.1/ CONVOLUTIONAL NEURAL NETWORK

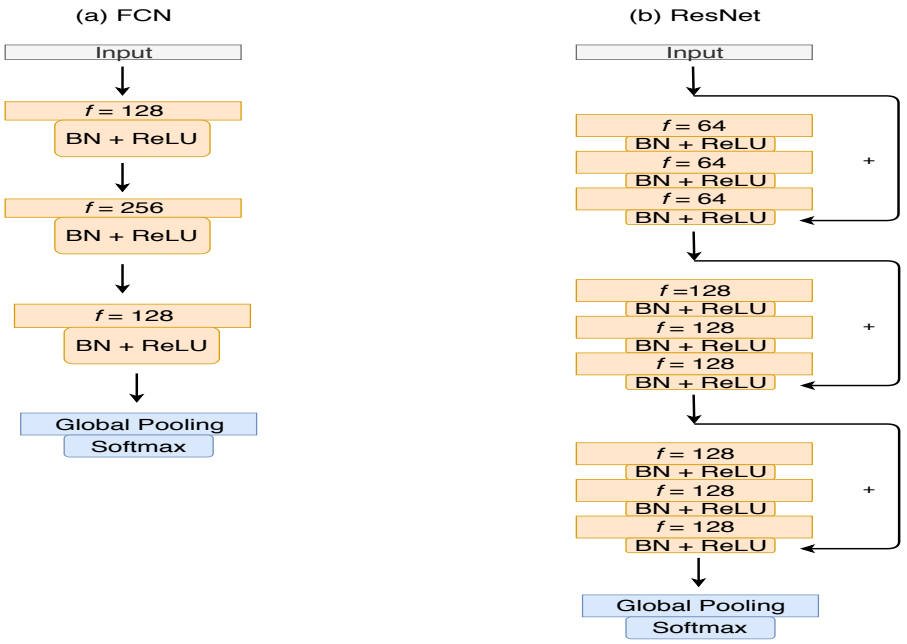


Figure 6.2: The network structure of FCN and ResNet tested in [164].

CNNs [5] are prevalent to image processing activities, reducing the number of parameters that need to be learned when the number of neuron connections in the hidden layer is limited to just certain input neurons (i.e. the local area in the image input). In many practical applications, CNNs have achieved positive results, particularly in the field of computer vision. In case of TSC problems, it is possible for CNNs to learn or automatically extract features from raw input data. A sequence of observations can be processed as a one-dimensional image, that can be read and distilled in the most salient elements by a CNN model.

This work considers two convolutional network architectures: Fully Convolutional Networks (FCN), and Residual Network (ResNet). Both of these architectures have been tested in [164] and attained premium performance to the state-of-the-art TSC approaches.

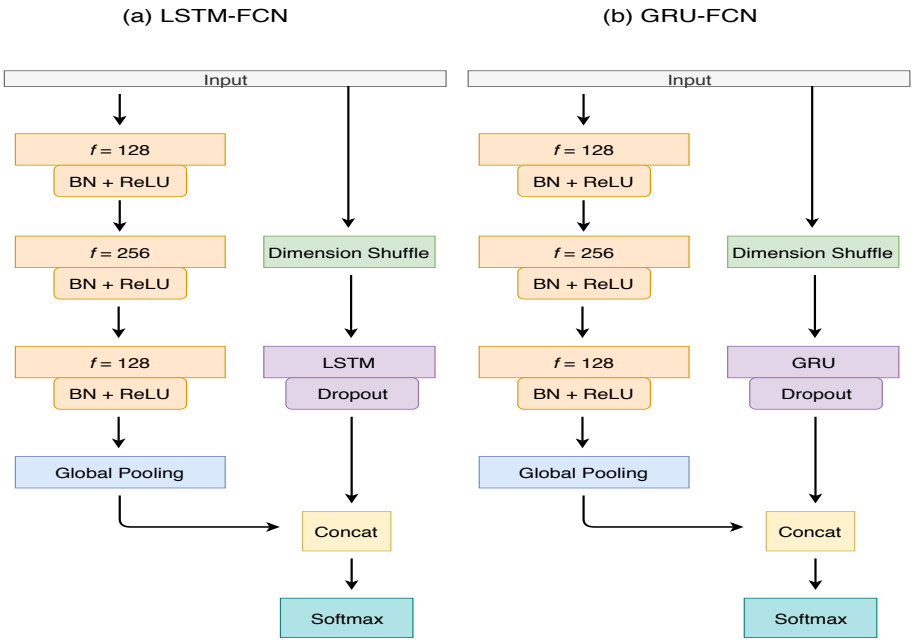


Figure 6.3: The network structure of LSTM-FCN and GRU-FCN proposed in [206] and [200] respectively.

For semantic segmentation on images, FCN [99] has demonstrated impressive quality and efficiency. The FCN used for TSC is constructed by stacking three blocks, each of which consists of a convolutional layer with f filters, followed by a batch normalization layer and a ReLU activation layer as shown in Figure 6.2-a. The features are fed into the global average pooling layer after the first three convolutional blocks, largely reducing the number of weights and the final result comes from the softmax layer.

ResNet is a deep CNN that uses shortcut connections between successive convolutional layers for training. The architecture used for TSC consists of 9 convolutional layers, followed by a general average pooling layer as shown in Figure 6.2-b.

6.3.2/ Recurrent Neural Network

RNN is a network that has loops. The Long Short-Term Memory, or LSTM, is a type of RNN. In the context of deep learning, LSTM has recently been widely used as it is free from the problem of vanishing gradients and offers excellent performances and results. LSTM based networks are suitable for classifying time sequences and replacing many conventional approaches to deep learning.

LSTM-FCN architecture has been proposed in [206]. The authors improved the previously discussed FCN's performance by transferring the time-series into a shuffle layer and subsequently passing it into a LSTM block as shown in Figure 6.3-a. The output of the LSTM block is concatenated with the output of the global pooling layer and the final results are obtained from the softmax layer.

Similarly, the authors in [200] proposed the GRU-FCN architecture, shown in Figure 6.3-b. The proposed architecture has replaced the LSTM with a Gated Recurrent Unit (GRU), which produces higher classification accuracy.

The application of deep learning models for TSC task will be further discussed in Chapter 8.

6.4/ IoT Time Series Compression

In order to study the impact of data compression on the performance of DNNs for univariate TSC, three compression techniques were considered in this work: Compressed Sensing (CS), Discrete Wavelet Transform (DWT), and an error-bounded lossy compressor, namely SZ. As SZ is described in Chapter 5, this section describes the CS and DWT techniques.

6.4.1/ Compressed sensing

Compressed sensing is a digital signal processing technique capable of effectively acquiring and rebuilding a signal from a smaller number of measurements. CS can capture and display sparse signals at a rate that is considerably lower than the one normally used in the Shannon sampling theorem.

The basic measurement model for compressed sensing is defined as in Equation 6.1:

$$[Y]_{M,1} = [\phi]_{M,N}[X]_{N,1}, \tag{6.1}$$

where X is the original sparse signal of length N, Y is the compressed signal of length $M(M << N)$, and $\phi \in \mathbb{R}^{M \times N}$ is the sensing matrix that enables the reconstruction of the signal.

Most signals generally have a low sparsity in the time-domain representation. The representations of these signals are therefore sparse in the frequency domain or under a certain basis. In that case, X can be represented using ψ as a sparse vector as shown in Equation 6.2:

$$[X]_{N,1} = [\psi]_{N,N}[z]_{N,1}, \tag{6.2}$$

where z is the frequency/spectral domain representation of the original signal X, and ψ is the matrix that converts the signal to the temporal-domain from the spectral-domain. The reconstruction of the signal allows z to be calculated from Y by combining Equation 6.1 and Equation 6.2 as follows:

$$[Y]_{M,1} = [\phi]_{M,N}[\psi]_{N,N}[z]_{N,1}. \tag{6.3}$$

Note that in this work, ψ is calculated by applying the inverse Discrete Cosine Transform upon the columns of the identity matrix. Additionally, the l_1-norm minimization [15], also known as Basis Pursuit, has been employed to recover the original signal X.

In this work, we assume that the data collected are time series of floating point numbers, so the compressed vector Y contains floating point data. In that case, to encode the resulting Y vector, an additional lossless entropy coder is required. Furthermore, we utilize the lossless FPZIP compressor proposed in [26] for the encoding step.

6.4.2/ Discrete Wavelet Transform

DWT has many advantageous properties that are valuable for time series data mining [55]. It transforms and analyzes a time series at multiple resolutions using the so-called wavelets.

The DWT allows a time-frequency representation of a signal. It splits the signal into low-frequency components (approximations) and high-frequency components (details) through the use of filters. The motivation for exploiting the DWT lies in the transformation of redundant samples in the temporal domain to decorrelated coefficients in the time-frequency domain, which allows the compaction of the original samples and their representation with fewer coefficients. Hence, this process facilitates the study of certain features of the original dataset. The application cases of the DWT for time series analysis are excessively various, for example, compression and noise filtering [192, 193].

In this work, the DWT is used for compression by maintaining only a signal's approximation data. These approximations are floating point values that need to be encoded, similar to the compressed sensing method, the FPZIP was considered.

One of the methods that can be used to compress and denoise a time series using DWT is to retain only the approximation data after performing the DWT, as a perfect reconstruction can be done using these approximations, while the details can be discarded and considered as noise. During the reconstruction, the details data can be set to zero.

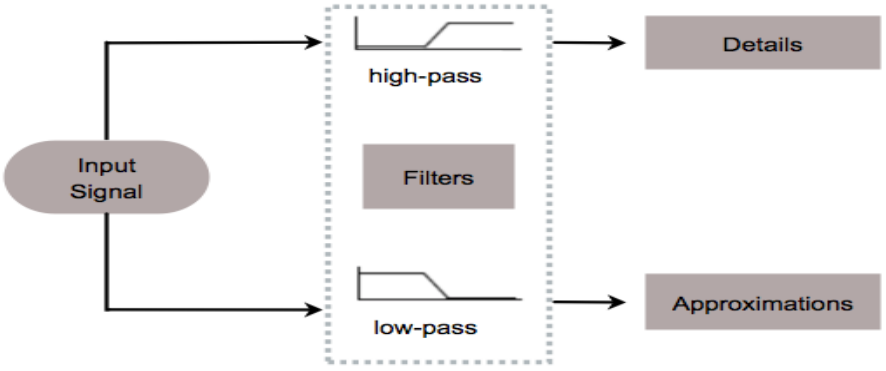


Figure 6.4: Discrete Wavelet Transform frequency portions of signal.

6.5/ PROPOSED COMPRESSION APPROACH

In this section, a compression scheme that enables a high compression ratio, denoises input data, and controls compression/decompression error is presented. It begins by presenting the lifting scheme implementation of the DWT and its capacity to compress and denoise the input signal, and then details the proposed compression scheme.

6.5.1/ FASTER IMPLEMENTATION OF THE DWT

The conventional method of wavelet transform needs complicated mathematical calculations, making it unfit for resource-constraining devices. Sweldens had proposed the lifting scheme in [6], a more efficient algorithm for calculating wavelet transforms and building bi-orthogonal wavelets.

The lifting scheme forward transform consists of three operations: split, predict, and update. The split operation divides the signal into even and odd samples. This step is also known as the lazy wavelet transform. In view of the locally correlated structure of the signal, even and odd samples are strongly correlated. The predict step predicts odd samples from even samples. Finally, The update step guarantees that the smoothed approximation signal has the same average as the initial signal. For a detailed explanation of the lifting procedure, refer to [192, 194].

There are many benefits of applying the lifting scheme to IoT devices: (1) It enables a faster implementation of the wavelet transform and is very appealing for real time applications with low energy. (2) It is memory efficient and enables a fully in-place calculation. (3) Very easy to understand, implement, deploy and adapt to various nodes.

6.5.2/ THE LIFTING SCHEME FOR NOISE FILTERING

The data collected from real world applications are usually noisy. Data reduction is the primary reason to implement the DWT. Equally important, eliminating the large variations of data mixed in the collected signal before transmission can be a crucial step in real-time applications as this step is no longer required at the edge of the network. Thus, taking less time to process and classify the transmitted data at the edge/sink.

Figure 6.5 shows an example of two level decomposition of an ECG signal. In the first decomposition, the initial signal is divided into approximations and details. The approximations maintain the information of the initial signal, and the details contain the fast varying changes that may be regarded as noise. The second decomposition reapplies the lifting forward transform on the approximations resulting in level two approximations with fewer data points and less noise.

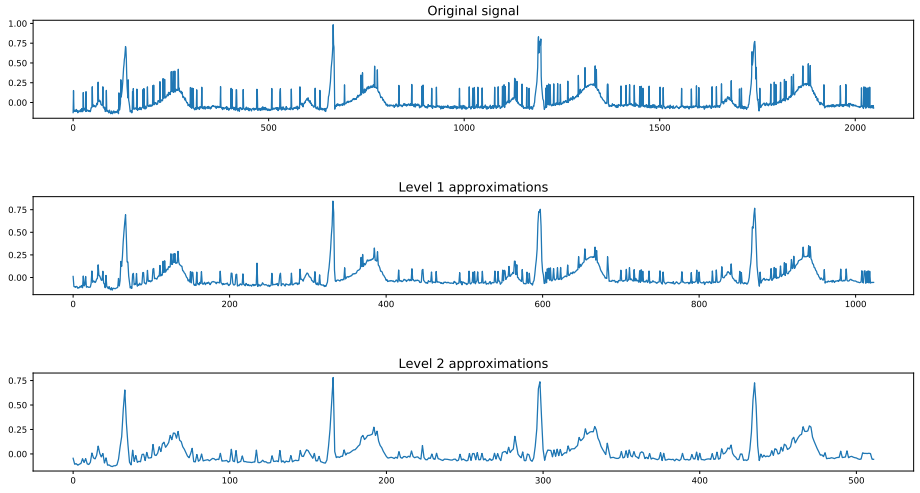


Figure 6.5: Example of two level decomposition of a noisy ECG signal.

In this work, we propose to apply the classification methods on the approximations of the original data. This is driven by the reality that approximations are smaller in size than the original raw data, leading in quicker implementation of the algorithms and reducing data access time. In addition, the DWT's noise filtering capacity isolates important features from the original signal. The meaningful data is thus fed to the deep learning model.

6.5.3/ PROPOSED COMPRESSION SCHEME

The data collected from sensors in IoT applications are generally multivariate time series. An M-dimensional multivariate time series, $X = [X_1, X_2, ..., X_M]$ comprises of M different

univariate time series. In our compression scheme, the first step is to transform each univariate time series into approximation and detail sets using the lifting scheme. This step is referred to as column-lifting. Then the details are removed, maintaining only an M-dimensional set of approximations $A = [A_1, A_2, ..., A_M]$. Note that the lifting scheme was implemented based on the Haar wavelet due to its simplicity. Multiple levels of transformation can be done by reapplying the wavelet function on the approximation coefficients. The number of levels can be defined by the user depending on the application and the collected data.

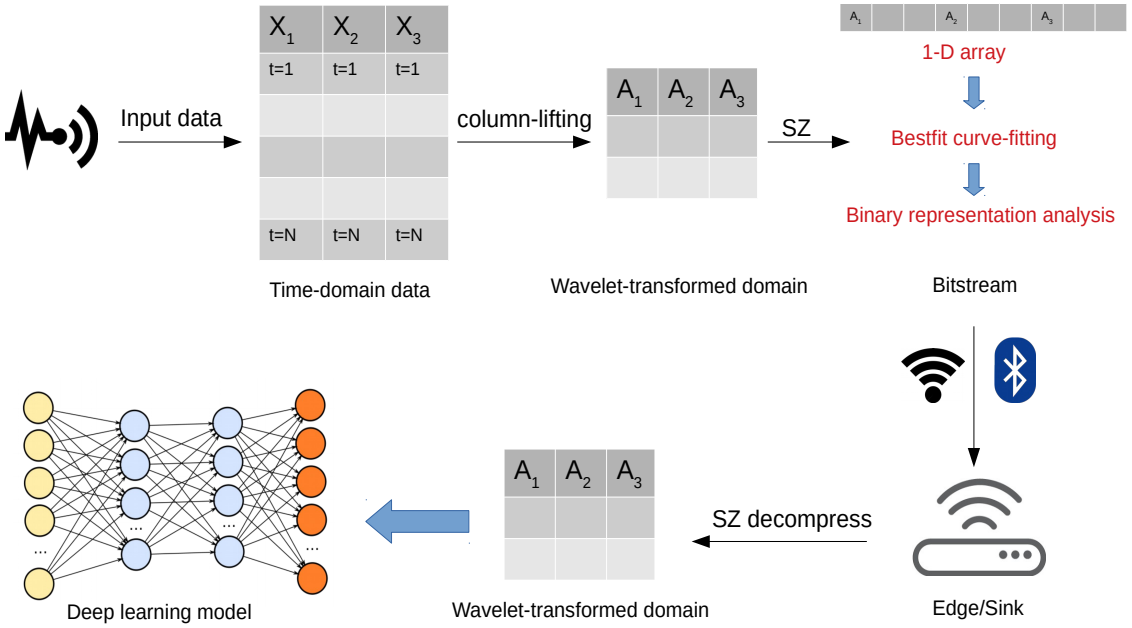


Figure 6.6: Proposed compression scheme for IoT time series classification.

Next, the SZ algorithm described in Chapter 5 processes the approximations array. The motive behind the combination of SZ and the lifting scheme is that SZ is a prediction method, and how smooth or jagged the time series affects its output. The benefit of the lifting scheme transformation is that it results in a smoother version of the initial time series that allows the SZ's bestfit curve-fitting to better predict the data in the user-required error. When the data reaches the edge/sink, the time series reconstruction method consists of using the SZ algorithm to decompress the compressed approximation coefficients. Then the wavelet-transform data is introduced into a deep learning model without the need to apply the inverse wavelet transform and recover the original data. This process is illustrated in Figure 6.6 and Algorithm 3. The input multidimensional array is firstly processed by the forward lifting transform. Based on the provided number of decomposition levels L, the three operations (split, predict, and update) are repeated L times. The results of this operation are one matrix of approximation coefficients A and a set of L matrices of detail coefficients D $\{d_1, ..., d_L\}$. The proposed approach consists of discarding the set D,

considered as noise, and keeping the approximations only. Then the approximations are converted to 1-D array, SZ's bestfit curve-fitting is applied and finally the unpredictable data points are represented using the IEEE 754 binary representation. The output is a binary array ready to be transmitted to the sink/edge.

Algorithm 3 Proposed compression approach

Require: input (N-D array), E (error bound), L (decomposition levels), M (num rows), N (num columns)
Ensure: $output$ (binary array)

```
output ← [ ]
//forward lifting transform
for 1 to L do
    split(input)
    predict(input)
    update(input)
end for
//get the approximations only
A ← approx(input)
// SZ compression
linearization(A)
Bestfit Curve-Fitting Compression(A, output)
Compressing Unpredictable Data(A,output)
```

6.5.4/ TIME COMPLEXITY AND COMPRESSION PERFORMANCE

The time complexity of the DWT lifting scheme is $\Theta(N \log N)$, and the time complexity of the SZ algorithm is $\Theta(N)$. Figure 6.7 shows the execution time in seconds of the DWT, SZ, and DWT+SZ methods written in C on a Polar M600 watch for different numbers of floating-point data. The results show that the execution time of the SZ technique is low, and the proposed technique needs around 0.02 Seconds to process 2000 data points, 0.04 Seconds to process 8200 data points, and around 0.13 Seconds to process 33000 data points.

In order to test how the use of the DWT can help increasing the compression ratio, 3-axis accelerometer data have been collected over 80 periods (1 period = 30 seconds). The user was at rest for the first 40 periods (sitting on a chair) and walking for the last 40 periods. Figure 6.8 shows the number of transmitted bytes after the compression of 36864 bytes of accelerometer data using SZ and DWT+SZ methods. The results show that the averages of the transmitted bytes per period are 742 bytes at rest and 1062 bytes at walking using SZ, and 446 bytes at rest and 718 bytes at walking using the proposed approach.

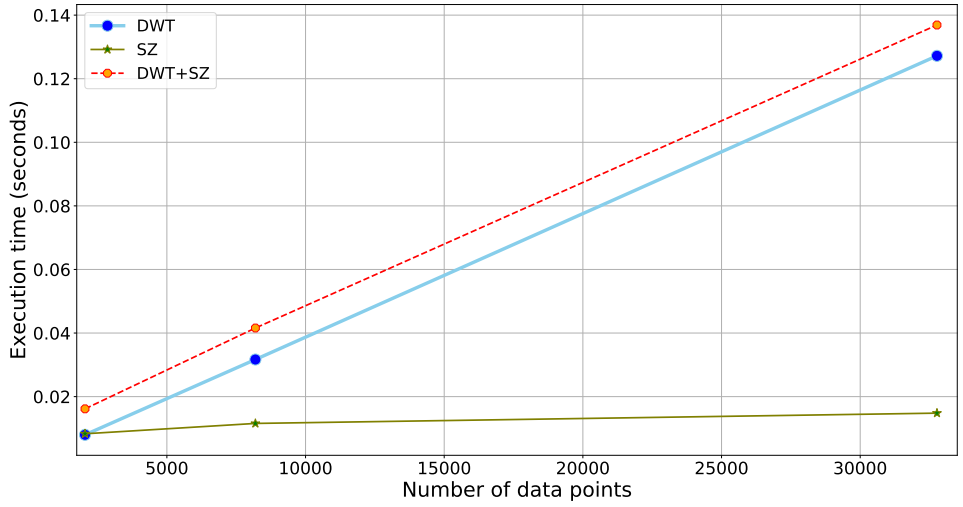


Figure 6.7: Execution time in seconds of the SZ, DWT, and the proposed compression scheme

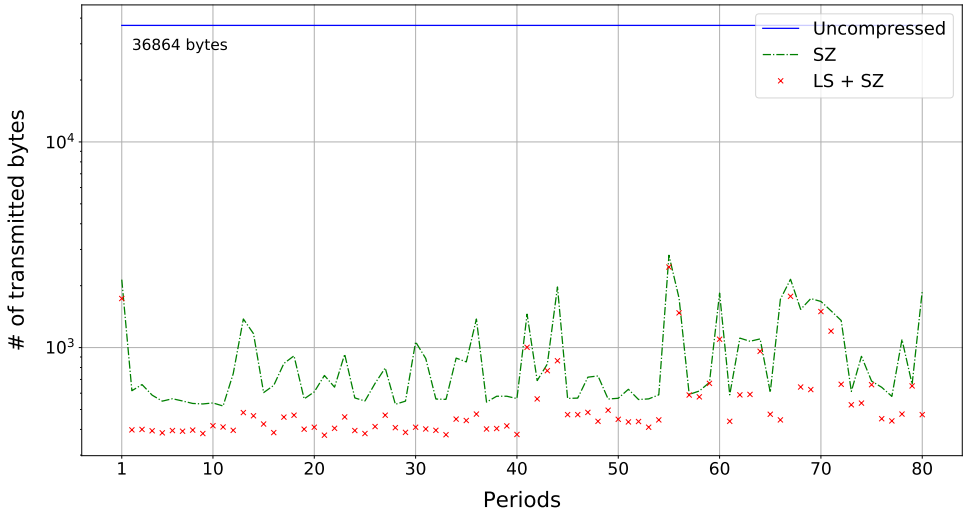


Figure 6.8: Compression of 36864 bytes of accelerometer data using SZ and DWT+SZ over 80 periods

6.6/ EXPERIMENTAL RESULTS AND ANALYSIS

The discussed compression techniques and classification models were tested on 8 UCR univariate time series datasets [113], 1 UCI multivariate time series dataset [72] and 3 UEA multivariate time series datasets [195]. Table 6.1 describes these time series and shows their source (ECG, Motion, Sensor, etc.), the size of the training and testing sets, the number of classification classes, the length of the time series and their type

Dataset	Type	# Classes	Length	Train size	Test size	# Features	Type
ECG5000	ECG	5	140	500	4500	1	U
Adiac	Sinusoidal	37	176	390	391	1	U
NonInvECGTh1	ECG	42	750	1800	1965	1	U
NonInvECGTh2	ECG	42	750	1800	1965	1	U
StarLightCurves	Sensor	3	1024	1000	8236	1	U
UWaveX	Motion	8	315	896	3582	1	U
UWaveY	Motion	8	315	896	3582	1	U
UWaveZ	Motion	8	315	896	3582	1	U
HAR [72]	Motion	6	128	7352	2947	9	M
ECG [195]	ECG	2	152	100	100	2	M
UWave [195]	Motion	8	315	200	4278	3	M
Wafer [195]	Motion	2	198	298	896	6	M

Table 6.1: The descriptions of the utilized datasets for time series classification based on [72], [195], and [113].

(U: univariate, M: multivariate), and the number of multivariate time series features. The results acquired on univariate and multivariate time series are presented in the followings of this section.

6.6.1/ SELECTION OF DECOMPOSITION LEVELS AND ERROR-BOUND

This section discusses the SZ error-bound value selection and the number of decomposition levels for the lifting scheme. The main goal of this work's study is to come out with an approach that considers the trade-off between the compression ratio and the classification performance. To answer this question, an empirical approach was taken, recording the model's classification accuracy trained on each dataset with different SZ error-bound and DWT decomposition level.

Figure 6.9 shows the outcome of the experiment on two datasets, namely Adiac and NonInvasiveFatalECG_Thorax1. For the SZ algorithm, the results show that the classification accuracy for an error bound of 10^{-3} remains very similar to that of the model trained on the original dataset, but the compression ratio is low compared to other compression settings. The highest compression ratio was achieved for an error bound of 10^{-1}, but the classification performance dropped significantly. Using a 10^{-2} error bound resulted in a good balance between classification accuracy and compression ratio. As for the number of decomposition levels, it can be seen that doing more than two wavelet decompositions greatly reduces classification accuracy while one level decomposition keeps the accuracy similar to the one of the original model and two levels decomposition helps to improve the compression ratio at the cost of slightly sacrificing the classification performance.

The proposed approach in this chapter is to combine the lifting scheme with SZ. Two combinations were considered in the followings according to the above experiment: DWT L1+SZ E-2 and DWT L2+SZ E-2.

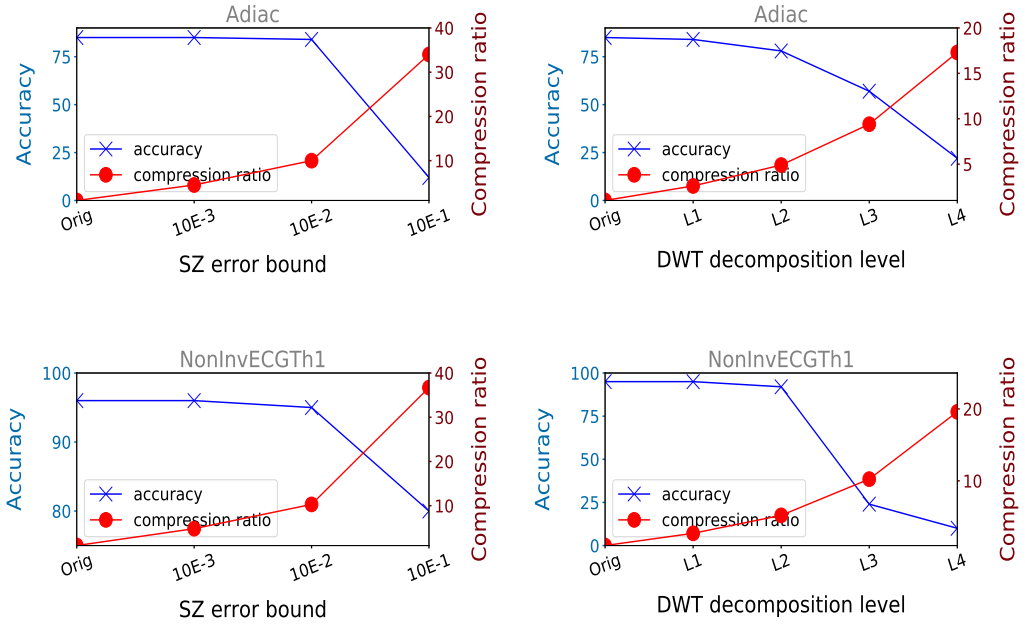


Figure 6.9: Accuracy vs compression ratio for different SZ error-bound values and DWT decomposition levels. The graphs correspond to Adiac and NonInvasiveFatalECG_Thorax1 datasets.

6.6.2/ UNIVARIATE TIME SERIES

The first studies were conducted on univariate time series. Two metrics were considered: compression ratio and classification performance.

6.6.2.1/ COMPRESSION RATIO

The following compares the compression ratios of the previously discussed techniques. Different variations of these techniques with different compression ratios were considered as follows:

- SZ E-1: SZ with an absolute error bound of 10^{-1} (the decompressed value should be within the range $[V - 10^{-1}, V + 10^{-1}]$)
- SZ E-2: SZ with an absolute error bound of 10^{-2}
- SZ E-3: SZ with an absolute error bound of 10^{-3}
- CS20: Compressed sensing with $M = \frac{N \times 20}{100}$
- CS40: Compressed sensing with $M = \frac{N \times 40}{100}$
- DWT L2: Two level DWT decomposition using Haar wavelet

- DWT L1: One level DWT decomposition using Haar wavelet

- DWT L1 + SZ E-2: One level DWT decomposition using the lifting scheme based Haar transform followed by SZ with an absolute error bound of 10^{-2}

- DWT L2 + SZ E-2: Two levels DWT decomposition using the lifting scheme based Haar transform followed by SZ with an absolute error bound of 10^{-2}

To compare the performance of the aforementioned techniques, the compression ratio (CR) is considered.

	ECG5000	**Adiac**	**NonInv-ECGTh1**	**NonInv-ECGTh2**	**StarLight-Curves**	**UWaveX**	**UWaveY**	**UWaveZ**
SZ E-1	14.73	34.37	36.73	37.91	69.82	32.97	35.07	33.97
SZ E-2	5.83	10.18	10.33	10.77	29.34	10.48	11.40	10.91
SZ E-3	3.07	4.58	4.81	4.90	10.85	5.61	6.12	5.63
CS40	3.12	3.2	3.43	3.46	3.73	4.11	4.34	4
CS20	6	6.12	6.62	6.68	7.14	7.57	7.97	7.54
DWT L2	4.8	4.9	5.28	5.33	5.71	5.31	5.39	5.31
DWT L1	2.52	2.58	2.74	2.77	2.99	2.92	2.98	2.91
DWT L1 + SZ E-2	9.06	13.09	13.95	14.48	29.89	13.23	13.93	13.40
DWT L2 + SZ E-2	12.3	18.8	21.2	21.7	32.7	19	19.7	19.1

Table 6.2: Compression ratios achieved by the different data compression techniques on the UCR univariate time series [113].

The obtained results are shown in Table 6.2. Table 6.2 shows the compression ratio of the different compression techniques. It can be seen that the best results are achieved by the SZ method. The closer the absolute error bound value is to zero, the higher the compression ratio is. As for the compressed sensing, the compression ratio depends on M, and on the number of decomposition levels in the case of DWT. Note that CS and DWT outputs are encoded using FPZIP.

As for the proposed approach, it can be seen that combining the SZ method with an appropriate error-bound and the DWT helps to achieve a higher compression ratio. For instance, SZ with an absolute error of 10^{-2} achieves a compression ratio of 5:1 on the ECG5000 dataset, while the compression ratio has been increased to 9:1 and 12:1 when combining SZ with DWT with one level decomposition and two levels decomposition respectively. The reason behind the improvement in the performance resides in the fact that each decomposition level of the DWT results in a more smoothed and correlated approximation coefficients. As a result, the SZ computes the approximation coefficients instead of all the data points in a time series. Furthermore, the DWT removes the noises from a time series, allowing the SZ method to perform better and to produce better results.

	Resnet		LSTM FCN		GRU FCN		FCN	
	Epochs	Batch	Epochs	Batch	Epochs	Batch	Epochs	Batch
ECG5000	200	16	400	128	400	128	200	128
Adiac	1500	16	4500	128	4500	128	1500	128
NonInvECGTh1	1500	16	4500	128	4500	128	1500	128
NonInvECGTh2	1500	16	4500	128	4500	128	1500	128
StarLightCurves	500	16	1000	128	1000	128	500	128
UWaveX	750	16	2000	128	2000	128	750	16
UWaveY	750	16	2000	128	2000	128	750	16
UWaveZ	750	16	2000	128	2000	128	750	16

Table 6.3: Experimental adjustments used in the implementation of the four DNN models.

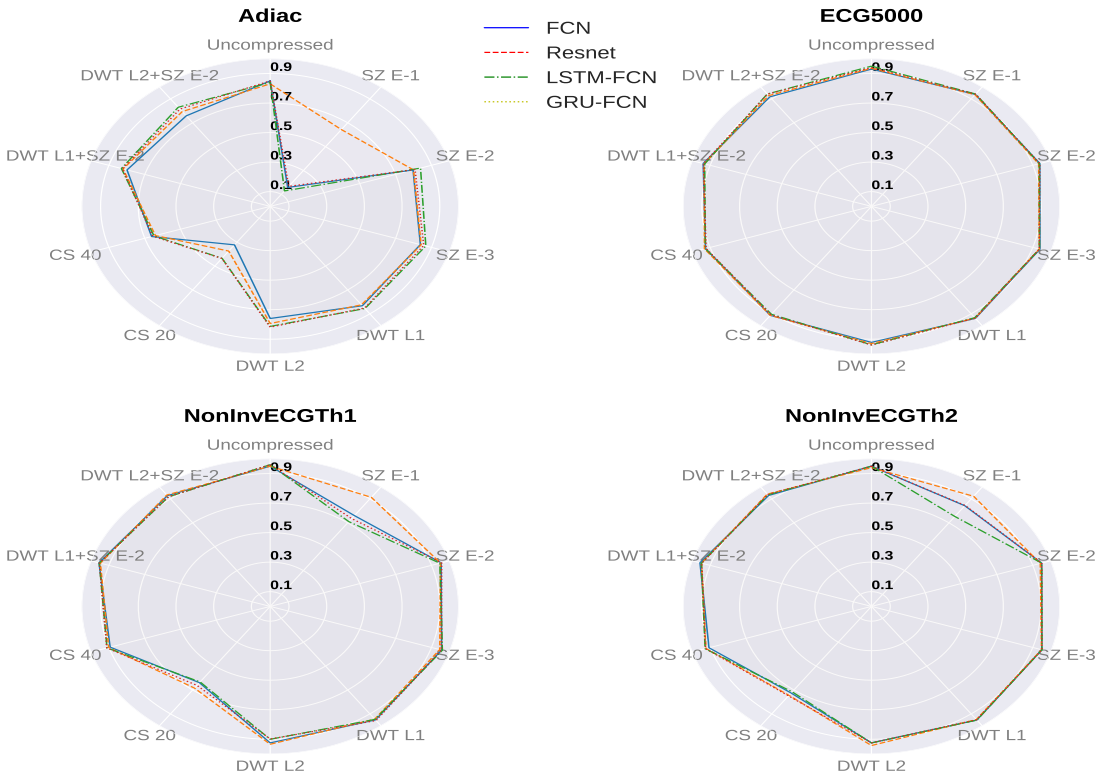


Figure 6.10: Accuracy metrics achieved by the four DNN models on the compressed periodic univariate time series.

6.6.2.2/ CLASSIFICATION PERFORMANCE

This section compares the four DNN models' classification performances on the compressed univariate time series. The considered experimental adjustments for these models are shown in Table 6.3.

In this work, the datasets used contain periodic and non-periodic time series. Periodic time series are like ECG and sinusoidal data, and non-periodic time series are like motion data. Each of the four DNN models was applied 5 times on the different compressed versions of the time series and the average accuracy was recorded. Figures 6.10 and 6.11

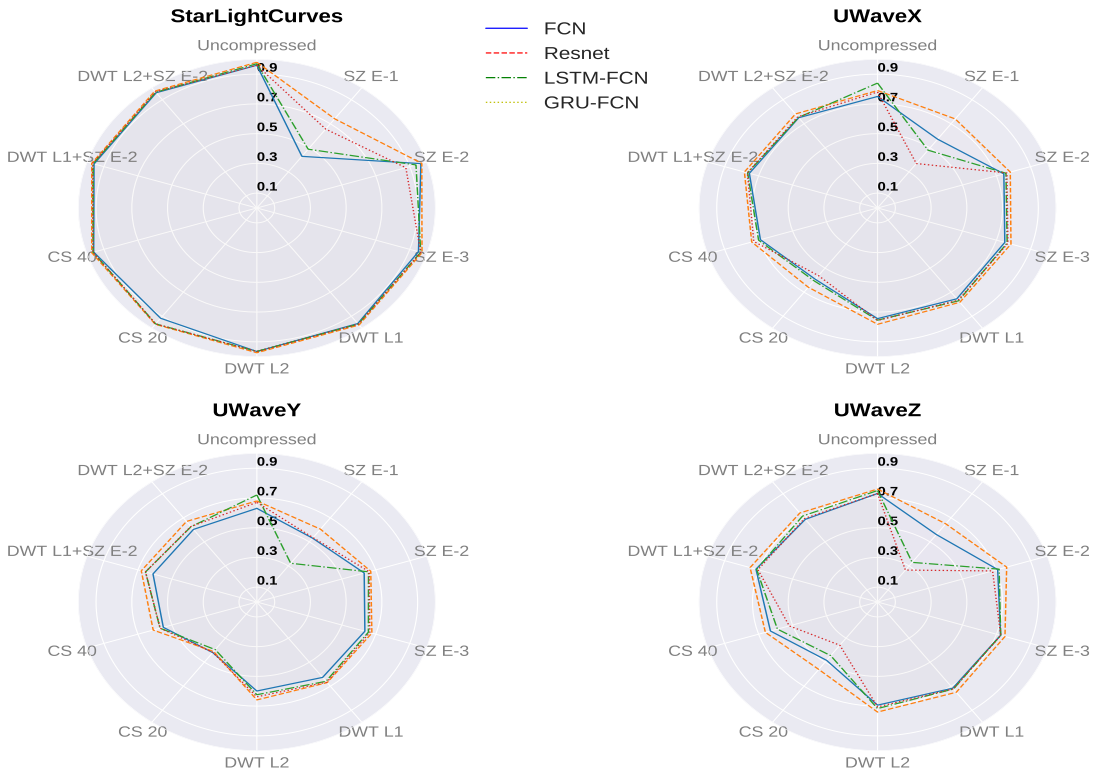


Figure 6.11: Accuracy metrics achieved by the four DNN models on the compressed non-periodic univariate time series.

represent each time series in a radar chart showing the accuracy metric obtained by each of the four NN models implemented with Keras on the compressed variants of the time series. Note that Figure 6.10 shows the periodic univariate time series and Figure 6.11 shows the non-periodic univariate time series.

It can be seen from these figures that the SZ method with an absolute error of 10^{-1} does not maintain the quality of the time series. Although it has the highest compression ratio, the classification accuracy of most datasets has been greatly affected. For example, the maximum classification accuracy values obtained on the uncompressed Adiac dataset and the compressed Adiac dataset with SZ E-1 (Figure 6.10) are 0.83 and 0.64 respectively. This can show the effect of data compression on the classification task. The SZ with absolute errors of 10^{-2} and 10^{-3} slightly affected the performance of the DNN models.

As for the CS method, the reconstruction quality of the compressed data was not satisfactory. In spite of the fact that the CS method is known to reconstruct a signal perfectly, the signal must be S-sparse, meaning noise-free, in order to be perfectly reconstructed, which is not the case for the time series used and the most real life data collected from IoT devices. This explains the performance decrease of the classifiers when working on reconstructed time series using CS. It can be noticed that the classification performance of the DNN models is highly affected by parameter M. For example, on the uncompressed

Adiac dataset, the maximum accuracy achieved is 0.84 versus 0.66 on the reconstructed dataset from 40% of the original data points (CS 40) and 0.4 on the reconstructed dataset from 20% of the original data points (CS 20).

On the other hand, the classification performance of the models was slightly affected by the DWT method, which was able to maintain the quality of the data. This can be explained by the denoising ability of the DWT method. The results also show that the classification accuracy values of the time series compressed with the proposed approach (DWT L1 + SZ E-2 and DWT L2 + SZ E-2) remained very close to those of the uncompressed time series. Taking into account the trade-off between compression ratio and classification performance, the proposed approach has yielded the best results on the univariate time series. It is important to note that the performance of the Resnet model was the most robust against compression in the experiments conducted. This can be seen in Figures 6.10 and 6.11, where data is compressed with SZ E-1 and CS 20, Resnet has produced the best accuracy compared to other models.

6.6.3/ MULTIVARIATE TIME SERIES

The compression approach presented can handle both univariate time series and multivariate time series. The followings address the effectiveness of the proposed approach when dealing with multivariate time series in terms of compression ratio, classification performance, and energy efficiency.

6.6.3.1/ COMPRESSION RATIO

The compression ratio of the different compression techniques achieved on the 3 UEA multivariate time series datasets in addition to the reconstruction time in seconds are shown in Table 6.4. Compared to the results obtained on the univariate time series, SZ with 10^{-1} error-bound and the two combinations of DWT and SZ achieved the highest compression ratio. However, in the case of multivariate time series, DWT L2+SZ E-2 outperformed SZ E-1 for UWave and Wafer datasets, and the compression performance of DWT L1+SZ E-2 became closer to that of SZ E-1. As for the reconstruction time, for the compressed sensing technique, the time needed to reconstruct the compressed data increases with the number of features in a dataset. This can be explained by the fact that CS uses a method of optimization to recover the data, and the more features a dataset has, the greater the process of minimizing the $l1$-norm. The SZ method's reconstruction time ranged from 0.02 to 0.05 seconds, making it very suitable for applications requiring quick decision-making. As for the DWT, the reconstruction time was not calculated since this work's proposed solution is to use the approximation coefficients directly for classification without reconstruction of the original data.

	ECG		UWave		Wafer	
	CR	rec time (sec)	CR	rec time (sec)	CR	rec time (sec)
SZ E-1	27	0.02	6.46	0.04	12.9	0.05
SZ E-2	5.4	0.02	3.36	0.03	4.4	0.05
SZ E-3	2.5	0.02	2	0.03	2.66	0.05
CS 40	2.8	15.4	2.7	255	2.64	280.2
CS 20	5.35	27.4	5.42	300.6	5.29	308.7
DWT L1	4.6	-	5.37	-	5.4	-
DWT L2	2.4	-	3	-	2.9	-
DWT L1+SZ E-2	12.6	0.02	6	0.03	8.37	0.05
DWT L2+SZ E-2	19.6	0.02	11.2	0.03	15.2	0.05

Table 6.4: Compression ratios and reconstruction time (seconds) achieved by the different data compression techniques on the UEA multivariate time series [195].

Another experiment was carried out in which 300 batches of 1000 feature vectors were taken from the HAR [72] dataset, each containing 9 features, and deployed on a Polar M600 wearable. We considered that each batch corresponds to a period of time. The techniques described in the previous section compress the corresponding batch for each period and the resulting number of bytes is recorded. After 300 periods, the average number of output bytes for each compression method is calculated with the standard deviation. The obtained results are shown in Figure 6.12.

Figure 6.12 indicates two main things: (1) the combination of DWT and SZ improved the compression performance, (2) the standard deviation of the number of bytes after compression at each period is high for SZ relative to CS and DWT. The latter can be explained by the fact that the compressed data are motion data (accelerometer and gyroscope), and these time series are generally jagged and vary rapidly between different activities (sitting, walking, running, etc). Since SZ is based on a prediction mechanism, its performance depends heavily on the characteristics and variability of the data. For the same reason, using DWT increased the performance of SZ due to each level of decomposition resulting in a smoother version of the data.

6.6.3.2/ CLASSIFICATION PERFORMANCE

The FCN model was used to evaluate the effectiveness of our compression approach on the multivariate datasets. The data processed by the model are tensors with the dimensions (samples, seq_len, n_channels), where samples is the number of training examples, seq_len is the number of steps in the time series, and n_channels is the number of columns (features) in the data.

The FCN model was trained on three variants of the data. In the first one, the model

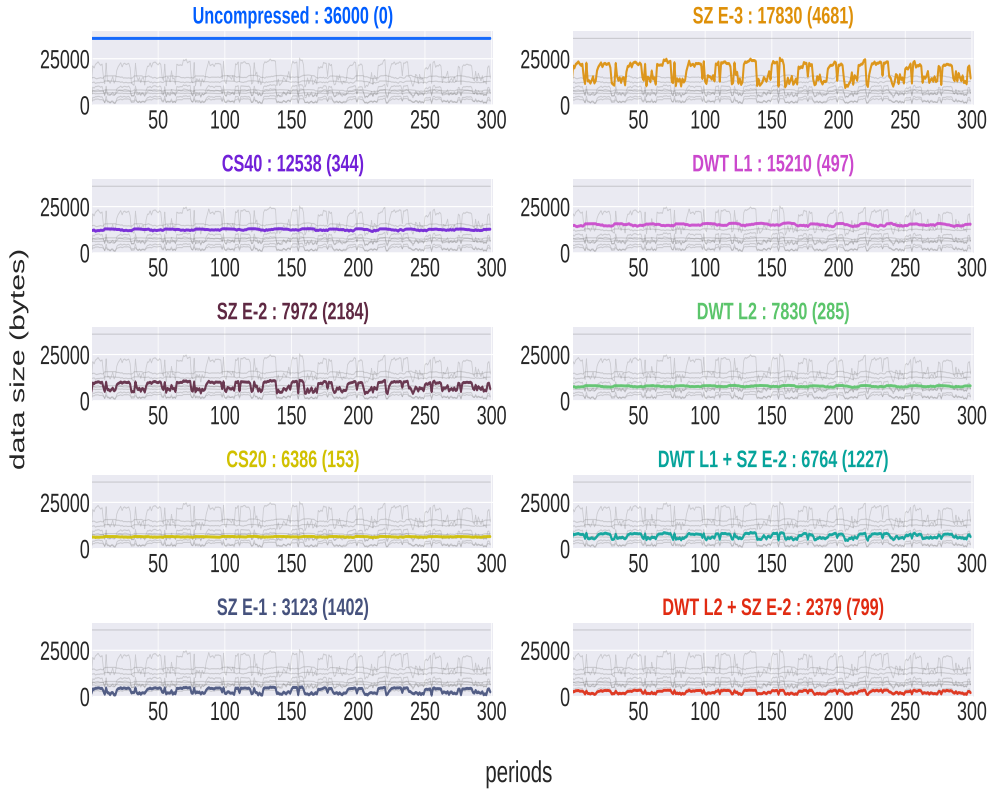


Figure 6.12: Compression of $[1000 \times 9]$ batches of the HAR [72] dataset over 300 periods. Next to each compression technique is shown the average number of bytes per period alongside the standard deviation (shown in parenthesis).

processes the original data. In the second and third variants, the model processes the approximations resulting from one level decomposition ($seq\_len = \frac{length}{2}$) and two levels decomposition ($seq\_len = \frac{length}{4}$) using the column lifting. Note that the length of each dataset is shown in Table 6.1.

Table 6.5 shows the classification accuracy achieved by the FCN on the four datasets with different input data. The results show that the model was able to learn from compressed data. The results also show that training on level-2 approximations decreased classification performance compared to training on level-1 approximations for ECG, UWave, and Wafer datasets. The model, on the other hand, learned well from the HAR dataset's level-2 approximations. The amount of training data will explain this discrepancy. The training size of the ECG, UWave, and Wafer datasets ranges from 100 to 300, while the training size of the HAR dataset is 7352, as shown in Table 6.1. The small amount of training data did not help the model to learn from level-2 approximations the necessary structure.

Dataset	Input data	Length	Accuracy
ECG	original	152	0.87
	level 1 approximations	76	0.86
	level 2 approximations	38	0.72
UWave	original	315	0.93
	level 1 approximations	157	0.93
	level 2 approximations	78	0.80
Wafer	original	198	0.98
	level 1 approximations	99	0.98
	level 2 approximations	49	0.89
HAR	original	128	0.93
	level 1 approximations	64	0.93
	level 2 approximations	32	0.92

Table 6.5: Classification accuracy achieved by the FCN model on the multivariate time series datasets.

6.6.3.3/ ENERGY EFFICIENCY

The proposed compression scheme was implemented on a Polar M600 wearable using Android NDK toolset in order to test the impact of data compression on energy conservation. The collected data are multivariate time series consisting of 3-axis accelerometer. These time series are transmitted to a local PC using Android BLE after each period. Figure 6.13 shows the wearable battery level change over 120 periods (1 period = 1 min) for four distinct situations:

- Yellow line: the wearable device is in the idle state.
- Red line: the wearable device is continuously collecting data (motion sensors are turned on).
- Green line: The wearable device is continuously collecting data and compresses and transmits data after each period.
- Blue line: the wearable device is continuously collecting data, and transmits data after each period (no compression).

The results show that the battery level decreased to 76% when continuously collecting data, to 72% when performing sensing, compression, and transmission, and to 56% when performing sensing and transmission. Note that we set the sensing frequency for testing purposes to the maximum in this experiment, generally gathering data with the default sensing frequency does not consume so much energy. From these outcomes, it can be seen that by implementing the proposed compression scheme, the device's lifetime can be improved by 16% after 2 hours.

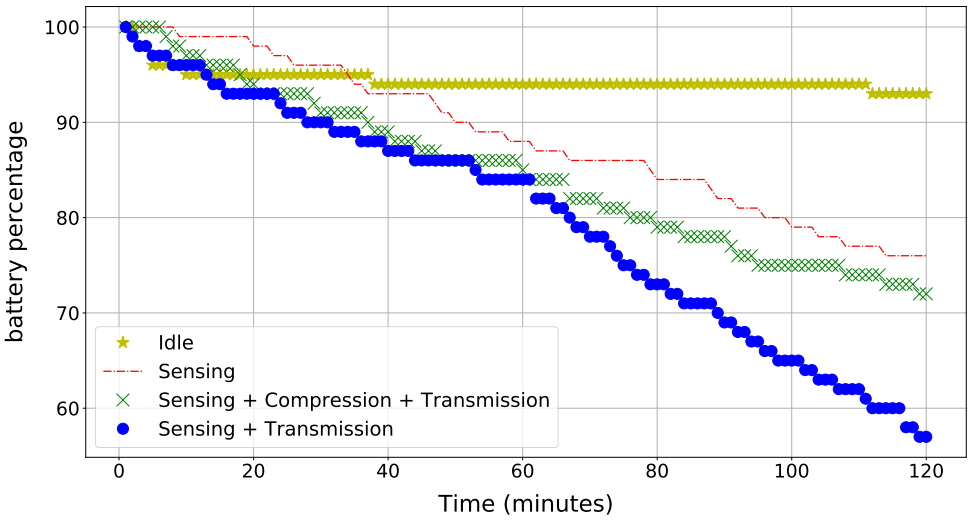


Figure 6.13: Polar M600 battery level over 120 periods for the four situations: Idle, Sensing, Sensing+compression+transmission, and sensing+transmission

6.7/ DISCUSSION

Various insights can be taken on the basis of the experiments conducted in this work. Recall that the main goal is to propose a compression approach that will help maximize an IoT device's lifetime. It operates on multivariate time series and floating point data efficiently. It can be tailored to any application, sensor or activity and suitable for near real-time applications.

The compressed sensing method with a low value of M showed a good compression ratio but after reconstruction did not maintain the required data features, thus decreasing the quality of the classification. Another weak point for compressed sensing is the reconstruction time, which is highly dependent on the data size and the number of features. Note that compressed sensing works perfectly when the sparsity of the data is high in the time/frequency domain, which is not always the case in IoT applications. On the other hand, the SZ method yielded a high compression ratio and the classification performance remained intact with the appropriate error-bound. One concern about SZ is that its performance depends on the stationarity of the time series as shown in Figure 6.12.

The proposed approach of using the discrete wavelet transform as a denoising and smoothing step prior to SZ can be useful for applications where the time series collected are jagged, such as recognition of human activity where high variations can be found between consecutive data points. The proposed approach helped to increase the compression ratio and maintain classification performance when using DWT with one level decomposition. Training a neural network with two levels decomposition approximations

decreased the classification accuracy by up to 13% for some datasets. Two things can be recommended to solve this problem: (1) Increasing the size of the training set or testing this approach on datasets such as HAR [113] which contain a good amount of training data enabling the model to learn from the compressed time series the necessary features. (2) Test this approach with wavelets that are more complex than the Haar wavelet. The selection of the corresponding wavelet family is considered for future work.

6.8/ CONCLUSION

Considering that IoT devices are generally limited in calculation, storage and energy, data compression can be seen as an effective way to increase the lifetime of these devices. Given the limitations of using a maximum compression ratio as stated in Chapter 5, an empirical study has been conducted in this chapter to identify appropriate compression ratios, which maintain important information from different time series. Different variations of compression techniques were considered based on an error-bound lossy compressor, namely SZ, the Discrete Wavelet Transform (DWT), and the Compressed Sensing (CS). Additionally, a new compression approach that combines the DWT lifting scheme and SZ has been proposed. The proposed method can manage univariate and multivariate time series effectively, attain a high compression ratio, and can be easily implemented and tailored to various applications.

In order to study the impact of time series compression on the classification task, four newly proposed deep neural networks for time series classification (TSC) were implemented. The experimental results showed that using DWT as a transformation step followed by SZ helps to denoise input data and enables SZ to compress more smooth approximation coefficients resulting in a higher compression ratio.

In addition, this chapter suggested to use the frequency-domain coefficients as an entry to the classification models, showing that deep learning models can still learn features from compressed data and attain good classification accuracy.

7

Deep Learning for Time Series Reconstruction and Artifacts Filtering

Recently, the need for fast, cost-effective, convenient and non-invasive cardiovascular analysis techniques is the primary attractive reason to use raw physiological signals such as electrocardiogram (ECG) and photoplethysmogram (PPG). Most wearable devices on the market today, are capable of collecting PPG data and enabling the measurement of important features such as heart rate, respiration rate, and blood pressure in addition to detecting irregular pulses and cardiovascular diseases. One major drawback of these raw data is their high sensitivity to motion, resulting in distorted signal and meaningless data zones. This chapter addresses two issues: (1) Resolving the limitations discussed in Chapter 5 of using a lossy compressor with high compression ratio, namely the distorted graphical representation and the inaccurate values of the extracted features due to signal deformation. (2) Proposing a neural network-based filtering method to remove corrupted zones from the collected PPG data in an unsupervised manner. It also proposes a PPG data summarization and augmentation strategy which optimizes the network performance.

7.1/ Introduction

Lossy compression techniques are known for achieving the highest reduction ratio, however, the high reduction in data size does not occur without expense, i.e, data distortion. This has been shown in Chapter 5, where the use of SZ with an absolute error-bound of 10^{-1} maintained the data quality of one dataset, and distorted the others. This inconsistency of the results obtained by maximizing the compression ratio led to the use of SZ with a lower error-bound such as 10^{-2} and 10^{-3} while combining it with a transformation

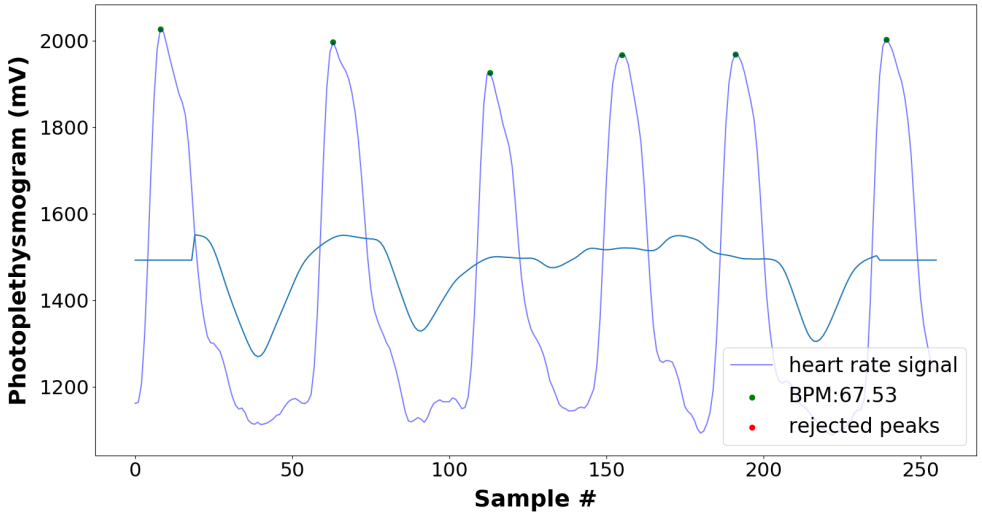
technique that removes the fast-varying changes, or noise, from the signal as shown in Chapter 6. The effect of data compression techniques on the data received for analysis is questionable when it comes to medical use, where the quality of the data is of great importance. The aim is not only to reduce energy consumption and data volume, but also to preserve data quality. Deep learning techniques have recently shown impressive results in extracting artifacts of compression from images. The first goal of the work done in this chapter is to apply this approach to physiological time series data in the same way by enabling better data recovery at the edge level using a 1-D convolutional autoencoder.

Given the general interest of this dissertation in addressing IoT-healthcare applications based on physiological signals and the specific interest of proposing solutions for stress detection applications, a dedicated emphasis will be provided to photoplethysmography in the second part of this chapter since this signal can be provided by most wearable devices today.

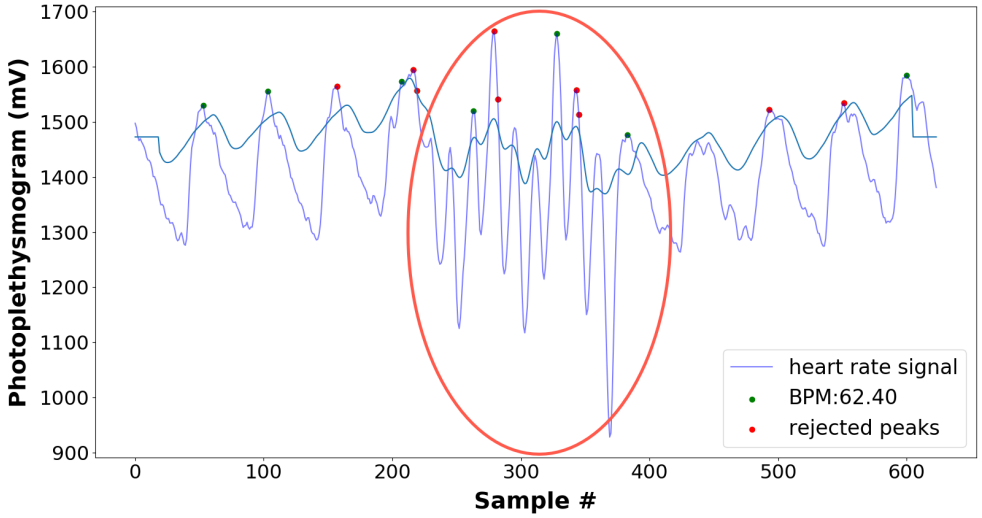
Wearable devices and fitness tracking applications have been introduced and used by millions of people, allowing continuous and unobtrusive tracking of an individual behavior and physiological features. One of the important signals to calculate the physiological features from these devices is the photoplethysmogram (PPG) signal. Photoplethysmography is a non-invasive optical method to monitor vital signs of the body such as heart rate, heart rate variability, and blood oxygenation. The PPG waveform represents variations in blood volume and contains important characteristics such as cycle period, baseline, and amplitude that are useful for analysis. Fortunately, wearable health monitoring devices, including smart watches and fitness trackers, can now capture PPG signals and allow cardiac activity monitoring by deriving from the PPG the same R-R intervals derived from the electrocardiogram (ECG).

Due to the enormous need to track chronic diseases and monitor elderly parents, ubiquitous health monitoring applications were listed among the various categories on the wearables market as the fastest growing segments [14][50]. Today, current wearable devices and sensors no longer focus on basic metrics for fitness tracking such as the number of steps taken, but also control essential physiological features. Commercial smartphone and wearable devices are currently able to measure a range of physiological parameters using PPG, such as the interval between successive heartbeats, respiration rate, and blood pressure [50].

One of the challenges of using PPG-based monitoring methods is the inaccuracy of the PPG signals during daily routine activities and physical exercises. This restriction is based on the fact that PPG signals are highly vulnerable to hand motion artifacts and environmental noise [197]. Examples of clean PPG and PPG with artifacts are shown in Figure 7.1. The estimation of heart rate variability data is strongly affected by artifacts, and the existence of these artifacts has consequences for anyone who relies on such data for



(a) clean PPG signal



(b) PPG signal with artifacts

Figure 7.1: Processing two PPG signals using the HeartPy library developed in [234].

higher-level analysis in addition to wasting storage space on meaningless data.

The aim of the second goal of this chapter is to address the artifacts in PPG data. The problem can be conceived as a problem of anomaly detection since the artifacts can be viewed in a PPG signal as anomalies. The difference between denoising a PPG signal and removing artifacts is important to note. An area containing artifacts such as in Figure 7.1(b) is an area where features such as heart rate variability and heart rate can not be extracted, while those features can still be extracted with confidence if the signal contains an acceptable noise level.

The ability to detect anomalies in a data stream is possible with the advances of deep learning and Neural Networks (NNs). Deep learning methods like Convolution Neural Network (CNN), autoencoders, and Long Short-Term Memory (LSTM) have been commonly used for anomaly detection problems [226]. This chapter proposes an unsupervised deep learning architecture based on a CNN-LSTM autoencoder that is able to detect artifacts in a PPG signal. Moreover, it proposes a sequences summarization approach for the neural network using the discrete wavelet transform (DWT) that enhances the training speed and and helps to avoid the problem of the vanishing gradients, as well as a data augmentation procedure for PPG data which enables a better generalization of the model.

7.2/ Related Work

7.2.1/ Data recovery and denoising

With the advances in deep learning and Neural Networks (NNs), the ability to recover or denoise a heavily distorted signal became possible. In [191], the author proposed a Deep Recurrent Denoising Neural Network (DRDNN) for electrocardiogram (ECG) signal denosing. The proposed model is a composite of a Recurrent Neural Network (RNN) and a Denoising AutoEncoder (DAE). The author used artificial ECG data to train the network and found that the network trained with artificial data performed better than the network trained with real data. In [165], the authors proposed a method for ECG signals denoising using a DAE and a wavelet transform method with soft thresholding. The primary concept is to use the wavelet transform to filter the basic noise of a signal and then use DAE to remove the complex residual noise that is difficult to remove using conventional techniques. The authors in [186] proposed the unsupervised Staired Multi-Timestep Denoising Autoencoder (SMTDAE) model to denoise gravitional waves containing non-Gaussian noise. The proposed model combines a DAE with a RNN architecture and is based on sequence to sequence bi-directional Long Short Term Memory (LSTM). By training the model on data with white Gaussian noise, the proposed model outperformed principal component analysis and dictionary learning in recovering signals with non-Gaussian noise. Since its emergence in 2014, the research community has developed considerable interest in the Generative Adversarial Network (GAN) models. The authors in [202] proposed the asymmetricGAN to remove the noise from EEG time series where the training data are unpaired, i.e, no clean versions of the noisy data are available. The proposed model is based on a generative adversarial architecture which does not require prior information about the source of the noise in the data. The results obtained were encouraging and demonstrate the potential medical use of GANs.

7.2.2/ PPG PROCESSING

Different research work in the literature tackled the processing and analysis of PPG data in the context of wearables and IoT. In [62], the technology of photoplethysmography and its potential applications was reviewed. This review focused on two important stages when dealing with PPG data, namely: pre-processing and feature extraction.

Van Gent et al did extensive work on analyzing and processing PPG. They developed the HeartPy library which works well with noisy PPG data [234]. The developed algorithm is capable of extracting the heart rate from the raw data and estimating the breathing rate. Additionally, they proposed a method that uses the R-R intervals and R-peaks to detect error/artifacts in the data [217]. In [214], the authors proposed a neural network model to enhance PPG measurements containing artifacts. The proposed approach consists of a global reference template reflecting a subject's clean heart beat morphology to be prestored in system memory. The reference template is derived through a beat quality assessment method from the acceptable quality beats of the current data. The extracted template should reflect the subject's clean heart beats morphology in order for the neural network to work well. In [42], the authors used frequency-domain analysis techniques such as fast Fourier transform and band-pass filtering to process PPG data. They referred to an adaptive echo cancellation method to remove the motion artifacts from the signal. The authors in [220] proposed to use the singular spectrum analysis and a spectral subtraction technique to reduce the corruption in the signal and remove the artifacts. The authors used the artifact-related 3-D accelerometer signal as an additional information to help eliminating the artifacts. Different other approaches have been proposed in [25, 45, 51, 69, 111] that use methods such as independent component analysis, Kalman smoother, wavelets, and adaptive filters in order to reduce the artifacts from PPG data.

The second contribution of this chapter, as compared to the above approaches, consists of detecting and removing the zones where the data are meaningless, and not denoising or enhancing the signal. The removal of these zones happens in an unsupervised manner without the need of extra information from other types of data or manually crafted features.

7.3/ BACKGROUND

7.3.1/ DISCRETE WAVELET TRANSFORM

The Discrete Wavelet Transform (DWT) allows a signal to be represented in a time-frequency domain. It divides the signal into components of low frequency (approximations) and components of high frequency (details) by using filters as shown in Chapter 6.

The inspiration to use the DWT in this chapter lives in transforming redundant samples in the temporal domain into decorrelated coefficients in the time-frequency domain, enabling the original samples to be compacted and represented with less coefficients [193, 194]. This process therefore helps facilitate the analysis of certain original dataset's features.

DWT has been used in several research works for time series classification tasks [13, 22]. Usually, accuracy is evaluated in accordance with other criteria when comparing several competing classification approaches. The classification algorithm's computation time (speed) is probably the second most important criterion, particularly for time series data. Since DWT produces a number of signal decompositions, the classification methods can be applied to the wavelet-transformed domain at a specific level. Compared to the original data, the wavelet coefficients are sets of smaller size, and therefore the computation speed of the classification method can be increased.

7.3.2/ AUTOENCODERS

An Autoencoder (AE) is an NN which uses data driven learning to extract features from a dataset. An Autoencoder has the same number of input and output nodes, and is trained to reconstruct the input vector instead of assigning a label to it. The addition of a noise in the encoding process is a successful strategy we can use to increase the robustness of the model. Denoising Autoencoder (DAE) can be seen as a regularization technique to address the issue of having more nodes in the hidden layer than in the input layer where the AE can simply copy those values without finding any meaningful features. As illustrated in Figure 7.2, the AE is trained on a randomly modified version of the input. Once the data is passed through the AE, the output is paired with the original values, not the modified values. The DAE is a stochastic version of an AE due to the stochastic corruption of the input while the uncorrupted version of the same input is used as the target for the decoding stage.

Besides being used as a method of regularization, the object of DAE is not to learn the exact representation of a sample in order to recreate it from low-demensional features, but rather to remove the noise from the input. DAE is now being used to reconstruct images from corrupted versions. This DAE functionality was successfully used for 1D signals as well such as for denoising time series. The first contribution of this chapter is heavily based on DAE's concept.

The autoencoder can be used for anomaly detection problems as well. This is done by learning the pattern of a normal process. A given input that does not follow this pattern is then categorized as an anomaly since the model will find it different from what it has learned during the training phase. The reconstruction error is the metric used to evaluate a given input. By defining a threshold, an input vector can be labeled as anomaly

if the difference between the values of this input and the output exceeds the threshold. Autoencoder-based anomaly detection can be used to process different types of data such as images and time series, therefore it can contains convolution layers (CNN autoencoder), long short-term memory layers (LSTM autoencoder), or a combination of both (CNN-LSTM autoencoder).

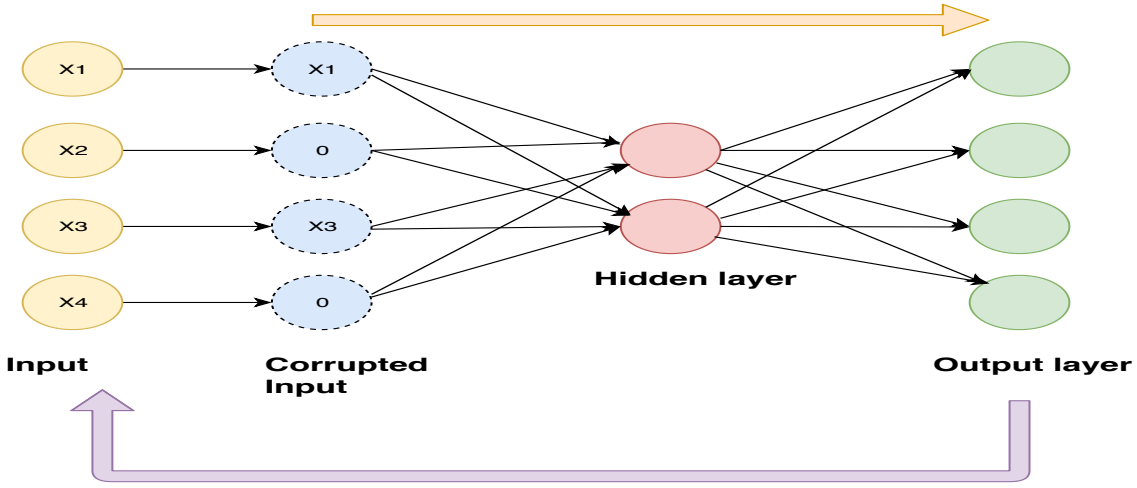


Figure 7.2: Schematic illustration of a simple Denoising Autoencoder.

7.4/ DATA RECOVERY AND ENHANCEMENT

This section addresses the limitations discussed in Chapter 5 by describing the proposed DNN model that allows the recovery of highly compressed data and presenting the experimental results.

7.4.1/ THE PROPOSED NETWORK ARCHITECTURE FOR DATA ENHANCEMENT

This section discusses the proposed technique used to extract the compression artifacts from the collected time series data. For this type of problems, it can be possible to have paired data. In other words, it is possible to have the uncompressed version of the reconstructed data in the data collection process to train the network.

The technique used to enhance the reconstructed time series is inspired from the work done in [199] for image denoising. The authors implemented a fully convolutional network consisting of an encoder decoder with skip connections. This architecture is based on the U-Net network originally presented in [128]. Even though the data collected from IoT are time series, we propose implementing a 1D U-Net with skip connections from encoder layers to decoder layers on the same level.

The proposed architecture takes a 1D signal input of size n as shown in Figure 7.3 and consists of three sections: the contraction, the bottleneck and the expansion section. The contraction part is made up of many blocks of contraction. Each block takes an input and applies two 1D convolution layers with a kernel size of 3 followed by batch normalization and 1D max pooling with pool size equal to 2. After each block the number of feature maps doubles so that the architecture can effectively learn the complex structures. The bottommost layer serves as an intermediary between the contraction layer and the expansion layer.

The expansion part also consists of several expansion blocks similar to the contraction layer. The decoding phase starts with the bottommost layer and an upsampling operation consisting of an upsampling layer with size of 2 followed by 1D convolution layer with a kernel size of 2 and batch normalization. Each expansion block connects the input by the corresponding contraction layer's feature maps and passes the concatenated result to two 1D convolution layers with kernel size of 3 followed by the upsampling operation. After each block, the number of feature maps get half to keep symmetry. This action will ensure that the learned features in the contraction part are used in the reconstruction of the time series. Note that the number of expansion blocks is equal to the number of contraction blocks. Given that the input values are in the range [0,1], the last top decoder layer is mapped back to the clean output signal with a 1D convolution with kernel size equal to 1 followed by a sigmoid activation. For the same reason, binary cross-entropy was used as the loss function [171] and variant of stochastic gradient descent 'Adam' [96] as optimizer.

7.4.2/ RESULTS

The first goal of this chapter is to enhance the data received to the edge/gateway using the proposed deep learning method. This dissertation considers IoT-healthcare applications context. Thus, medical data such as electrocardiogram (ECG) and photoplethysmogram (PPG) are used. The motivation behind the use of medical data is the importance of the quality of these data for analyzing and extracting meaningful information. That is, a high compression ratio applied to medical data may make the analysis unfeasible. This section shows how the proposed deep learning technique changes the quality of the reconstructed data from unacceptable to acceptable.

The ECG dataset used is the ECG-ID Database that was published in Physionet [9] and originally proposed in [241]. The PPG dataset was recorded in the Femto-ST laboratory department DISC, Belfort, France, using Shimmer3 GSR+Unit [237]. Note that these datasets were introduced and used in Chapter 5. The computations were performed on a NVIDIA Tesla Titan X GPU to train the NN on these datasets.

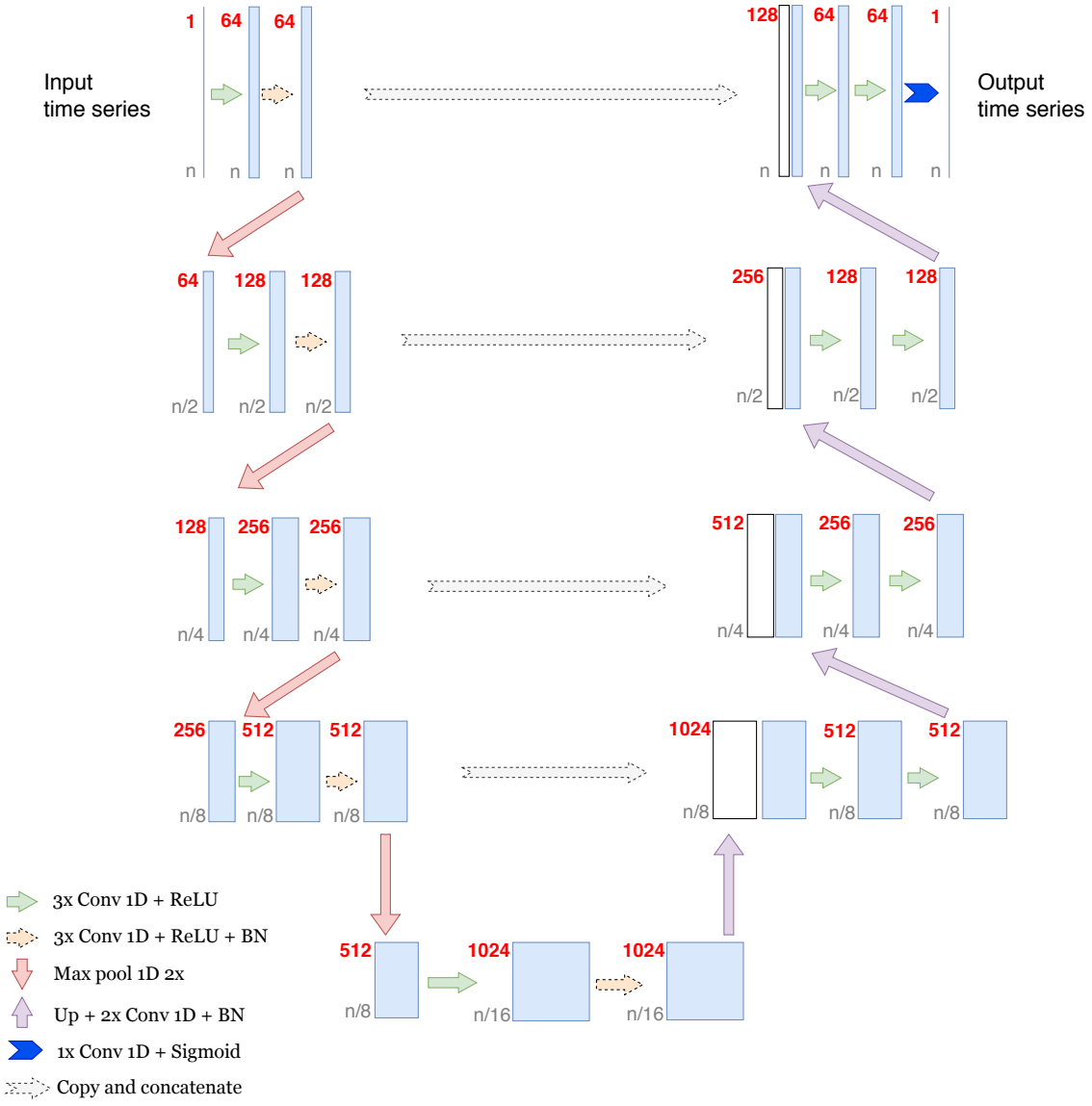


Figure 7.3: Schematic diagram of the proposed deep encoder-decoder network architecture.

This section illustrates how, when applied to highly compressed data, the proposed deep learning model in this section can improve the quality of the data and make it feasible for analysis. The standard evaluation strategy used for such problems is to compare the root mean square error of the reconstructed and enhanced data. It may not be enough for all applications, however, and especially for medical applications. Different features from the time-domain and frequency-domain of the reconstructed and improved signals are extracted in the use case considered and compared to those extracted from the original signals. Table 7.1 describes the features extracted from ECG and PPG signals by applying time and frequency domain analysis such as FIR-filters and fast fourier transform methods.

Metrics	Description
RMSE	Root mean square error
RMSSD	Root mean square of the Inter-beat (RR) Intervals (the time intervals between consecutive heart beats)
meanNN	Mean RR interval
BPM	Beats per minute
pNN20	The number of interval differences of successive RR intervals greater than 20 ms divided by the total number of RR intervals
pNN50	The number of interval differences of successive RR intervals greater than 50 ms divided by the total number of RR intervals
sdNN	Standard deviation RR interval

Table 7.1: Features extracted from the original, reconstructed, and enhanced PPG and ECG signals.

	RMSE	RMSSD	meanNN	BPM	pNN20	pNN50	sdNN
Original PPG		4.33	999.44	60	0	0	2.89
Reconstructed PPG	0.05	64.50	997.48	60	0.76	0.61	41.09
Enhanced PPG	0.02	8.86	1000	60	0	0	5.71
Original ECG		1.81	1000	60	0	0	1.09
Reconstructed ECG	0.05	116.51	974.06	68	77.27	63.63	85.28
Enhanced ECG	0.01	3.24	1000	60	0	0	1.73

Table 7.2: Comparison of RMSE and the features of the original, reconstructed and enhanced signals.

Table 7.2 shows the metrics taken from the original, reconstructed, and enhanced signals. Note that the average values of the features extracted from the original and reconstructed signals are the same as the ones shown in Chapter 5. The proposed NN reduced the RMSE for the PPG signal from 0.05 to 0.02 and the ECG signal from 0.05 to 0.01. More specifically, features such as RMSSD, pNN20, pNN50 and sdNN extracted from the reconstructed signal can be found to differ significantly from those extracted from the original signal. Having this high difference makes it impossible to analyze and study the signal. On the other hand, when the reconstructed signal was improved, these same features became very similar to those of the original signal. Figures 7.4 and 7.5 show different batches of the original, reconstructed, and enhanced signals taken from the testing set. It can be seen from these figures how a compression algorithm can highly degrade the shape of a signal, and how NNs can restore and enhance the shape, and allow the use of lossy compression techniques with a higher compression ratio for IoT-healthcare applications when the data quality is critical.

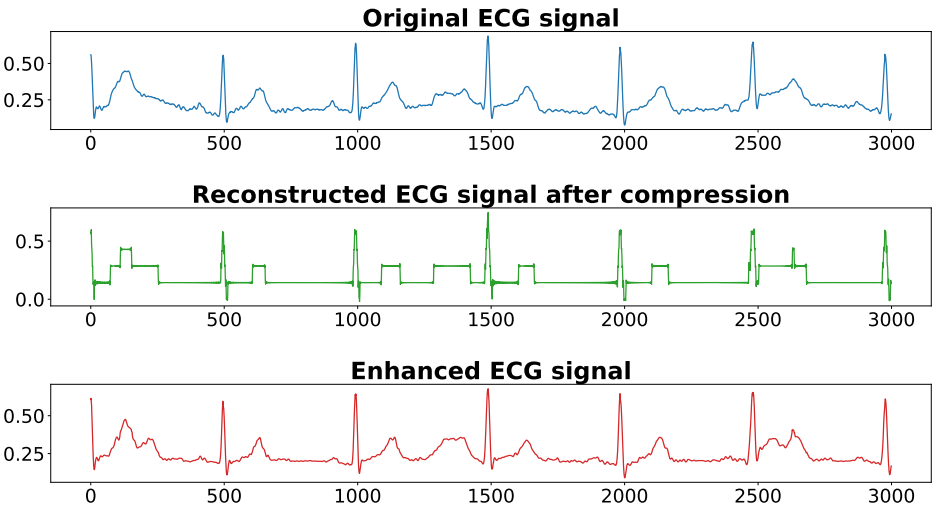


Figure 7.4: ECG signal enhancement. An example of a compression, reconstruction and enhancement of 3000 ECG data points.

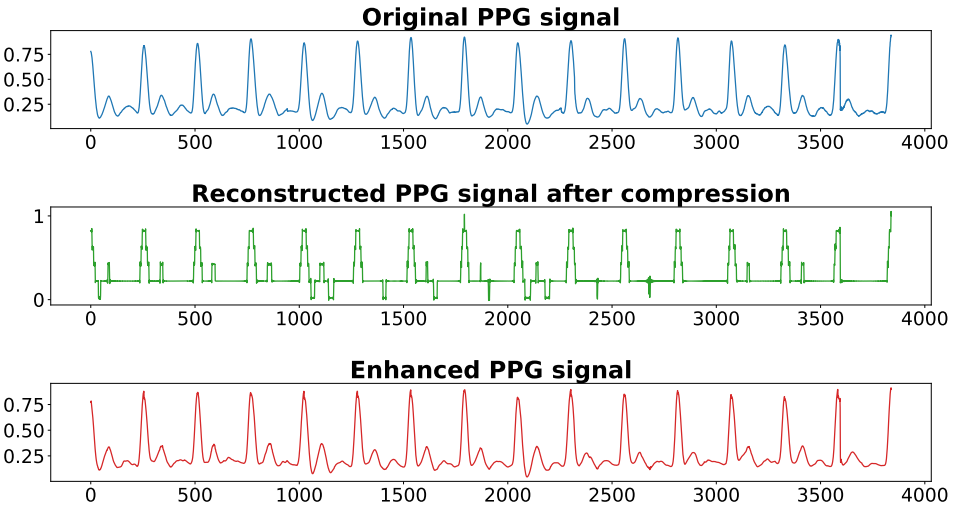


Figure 7.5: PPG signal enhancement. An example of a compression, reconstruction and enhancement of 3000 PPG data points.

7.5/ PPG ARTIFACTS FILTERING

This section presents a PPG data augmentation approach and then discusses the proposed DNN model that allows the filtering of the artifacts in a PPG signal.

7.5.1/ DATA AUGMENTATION

The approach to data augmentation used in this section is to apply various types of noise to the existing time series. The purpose of data augmentation is to prevent overfitting and enhance a deep learning model's generalization ability. Different approaches to data augmentation were proposed in the literature such as in [188]. While working with medical signals such as PPG, the main challenge is to ensure that the signal produced follows the temporal order of samples and the original signal form/shape. In other words, the peaks should still be easy to detect and the time between two consecutive peaks should stay the same so that the same features can be derived from the newly generated data as from the original data.

7.5.1.1/ GAUSSIAN AND UNIFORM NOISE

Uniform noise has a flat distribution which ranges between 0 and 1, which means that it is equally likely to draw all values between 0 and 1. Normally distributed noise, or Gaussian noise, has a zero-centered "bell-curve" distribution, with most values clustered to zero. Through adding and multiplying by some constants, the distribution of both uniform and Gaussian noises can be shifted and stretched. Such constants are carefully selected to keep the produced signal analyzable and meaningful and their values depend on the application and the available data.

These two types of noise are referred to as "white noise" since they have a flat power spectrum.

7.5.1.2/ SCALING AND MAGNITUDE-WARPING

The scaling approach adjusts the magnitude of the data in a batch by multiplying by a random scalar, while magnitude-warping adjusts the magnitude of each sample in this batch by convolving the data with a smooth curve that varies around one. These two approaches were inspired from the work proposed in [188]. The results of applying scaling and magnitude-warping to a PPG time series are shown in Figure 7.6.

7.5.1.3/ PINK AND BROWNIAN NOISE

Pink noise, also known as fractal noise or $\frac{1}{f}$ noise, has a power spectrum that decreases as the frequency increases. Pink noise is weighted toward low frequencies, as its power decreases like $\frac{1}{f}$. The application of a vanishing frequency filter is one way to compute pink noise [92].

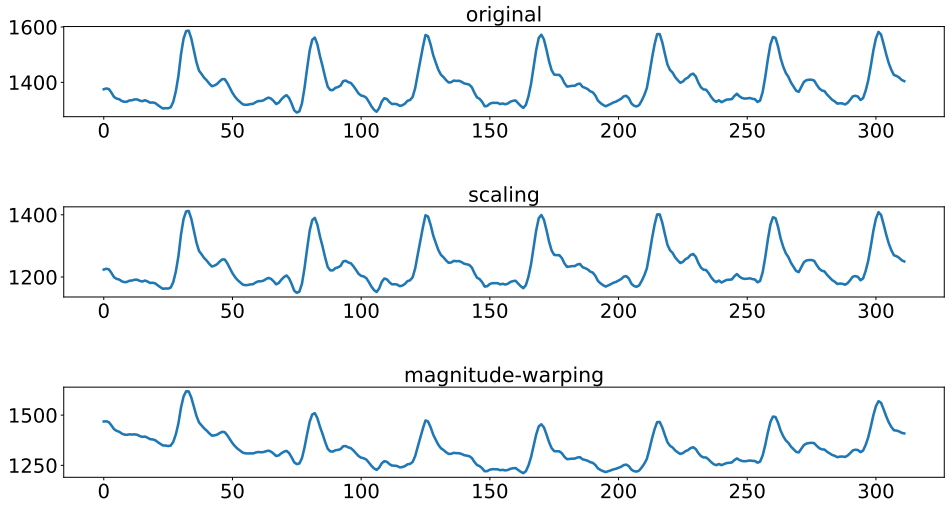


Figure 7.6: Applying scaling and magnitude-warping to a PPG time series.

Brownian noise further lowers the higher frequencies than the pink noise. It is also known as random walk noise and is calculated by integrating random noise that is normally distributed. Brownian noise has memory, which means that its past values influence each time point. Figure 7.7 shows the results of adding pink and brownian noise to a PPG time series. Although the shape of the time series has been affected, it is still possible to extract the relevant features.

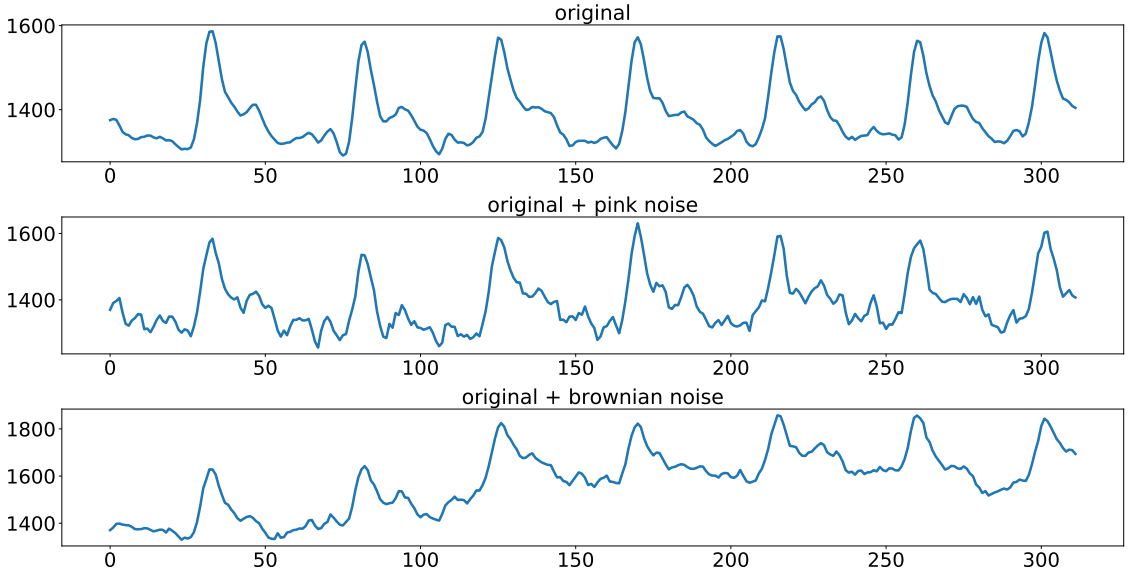


Figure 7.7: Adding pink and brownian noise to a PPG time series.

7.5.2/ PROPOSED NEURAL NETWORK MODEL

In this work, we propose the use a hybrid model, namely CNN-LSTM autoencoder, to detect the artifacts in a signal in an unsupervised manner. Given that PPG data are time series, an interesting approach is to use a model based on LSTM. If input sequences are very long, such as a PPG trace of hundreds of time steps, LSTMs can be difficult to use. The first problem of dealing with long sequences is the very long time required to train the model. Additionally, the back propagation through long input sequences can lead to vanishing gradients and, in turn, an unlearnable model. The first step in this section's proposed solution is to address the issue of long input sequences. Note that the goal is to check that a PPG data window is clean or contains artifacts, and that window can contain hundreds to thousands of data points. The approach taken is sequences summarization using the discrete wavelet transform (DWT). In natural language processing where input sequences are words, summarizing sequences were used, it might be possible to eliminate all words from input sequences above a defined word frequency. Instead of removing samples from the data, the use of the DWT will introduce a more compact version of the signal to the neural network. It can be viewed as a summary of the original signal with fewer data points. For example, if we take a PPG signal with length of 312 as shown in Figure 7.8, applying one-level decomposition will result in a set of 156 approximation coefficients and 156 detail coefficients. Then another decomposition can be applied to the approximations obtained, resulting in 78 approximation and detail coefficients. The model will then be trained on the approximation coefficients and the details will be discarded. As a result, the speed of the learning could be improved and the problem of the vanishing gradient avoided.

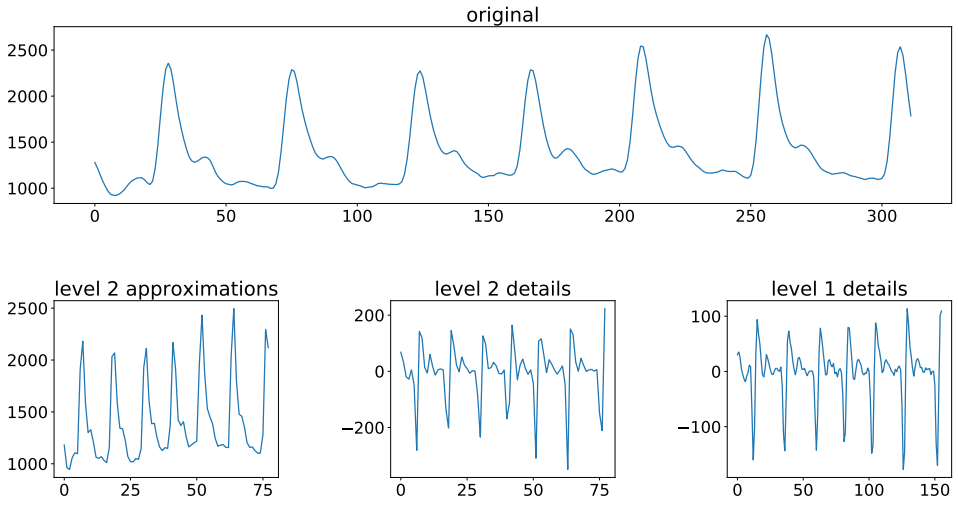


Figure 7.8: Example of the two-level decomposition of a PPG signal.

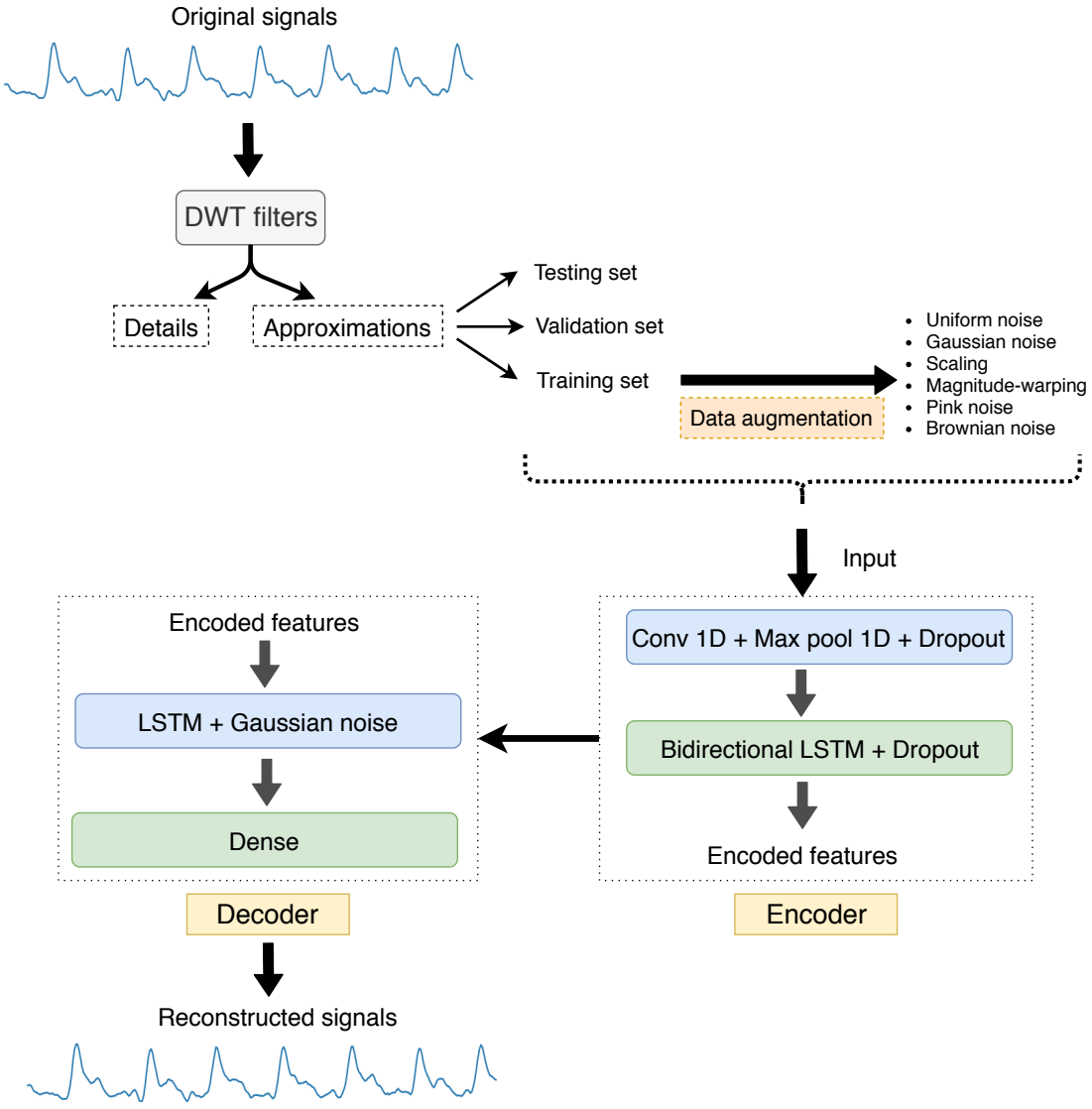


Figure 7.9: Proposed neural network approach for unsupervised PPG artifacts detection using CNN-LSTM Autoencoder.

The proposed architecture takes as inputs a fixed-length vector with shape $< S, T, 1 >$, where S is the number of data windows in a mini-batch, and T is the number of samples in each data window. Note that the data are standardized before being initialized to the model. The encoder part is made up of a 1D convolution layer with a kernel size of 5 and 320 feature maps, followed by a 1D max pooling layer with a pool size of 4 and a dropout layer with a rate of 40%. The output of the convolution layer is then flattened and fed into a bidirectional LSTM layer with 256 units followed by a second dropout layer. The decoder part consists of a LSTM layer with 192 units followed by a Gaussian noise layer for regularization and a time distributed dense layer. The linear activation function has been used for the output dense layer and ReLu [49] for the convolution and LSTM layers. Additionaly, the He normal initialization [118] was used for weights initialization,

the mean squared error was used as a loss function, and Adam [96] as optimizer with gradient clipping to avoid exploding gradients.

Figure 7.9 illustrates the proposed approach used for unsupervised PPG artifacts detection. Note that the data augmentation described in section 7.5.1 is applied on the training set after applying the wavelet transform on the original signals and splitting the resulting approximations into training, validation and testing sets. Then the model is trained only on clean PPG signals and tested on mixed PPG signals.

7.5.3/ RESULTS

The PPG dataset was collected using Shimmer3 GSR+Unit [237] with a sampling frequency of $Fs = 52$ Hz in the Femto-ST laboratory department DISC, Belfort, France. An overlapping sliding window was used to split the collected data set into batches, resulting in a training set containing 10212 batches of clean PPG and 3149 batches of artifacts, a validation set containing 4358 batches of clean PPG and 1369 batches of artifacts, and a test set containing 150 unlabeled windows of clean PPG and artifacts.

In the experimentation two strategies are considered. The first is to use PPG recording windows of 3 seconds ($Fs \times 3 = 156$ samples), so the model's input will be a tensor of shape $< S, 156, 1 >$. The second strategy is the proposed one, which takes larger PPG recording windows of 6 seconds (312 samples) and then applies two-level decomposition on these windows to reduce their length. As a result, the model's input will be a tensor of shape $< S, 78, 1 >$. The data augmentation process has been applied to the second strategy only in order to assess its advantage on the final output. The computations were performed on a NVIDIA Tesla Titan X GPU.

In order to determine what benefit training the model on wavelet approximations and augmenting the data give compared to the standard approach, we used the Precision vs. Recall and Receiver Operating Characteristic (ROC) curves. The ROC curve illustrates the trade-off between the true positive rate and the false positive rate for a model using different thresholds. Precision vs. Recall curve illustrates the trade-off between the true positive rate and the positive predictions for a model using different thresholds. Note that high precision is associated with a low false positive rate and high recall is associated with a low false negative rate. The thresholds in the curves are the autoencoder reconstruction error and the desired outcome is a model with high scores for both metrics (high precision and high recall).

Figure 7.10 shows that a better balance between recall and precision could be achieved by the proposed approach. If the model is trained on three seconds PPG windows with no data augmentation, the model will have a precision of less than 0.4 in order to achieve a recall greater than 0.9 while the model was able to have a precision of 0.9 with a

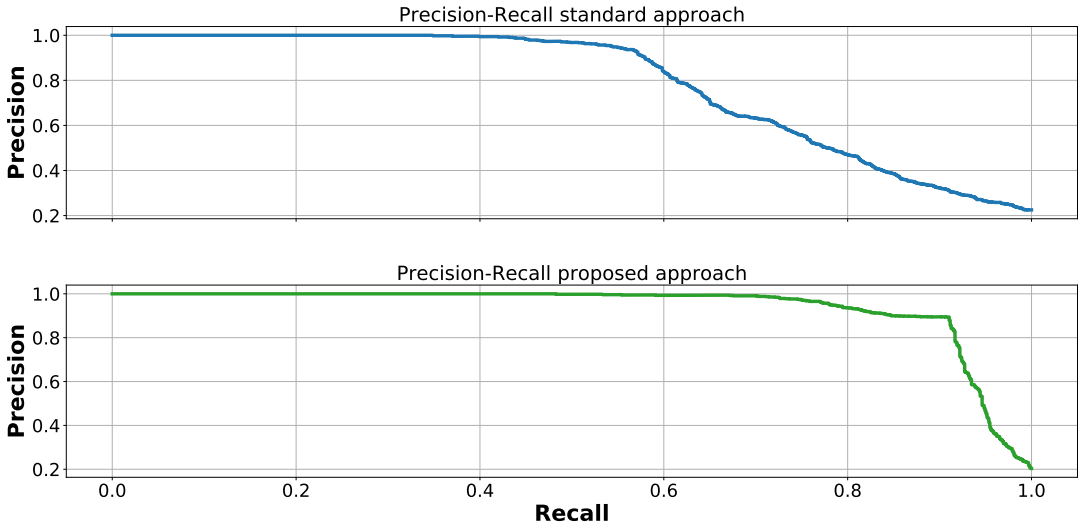


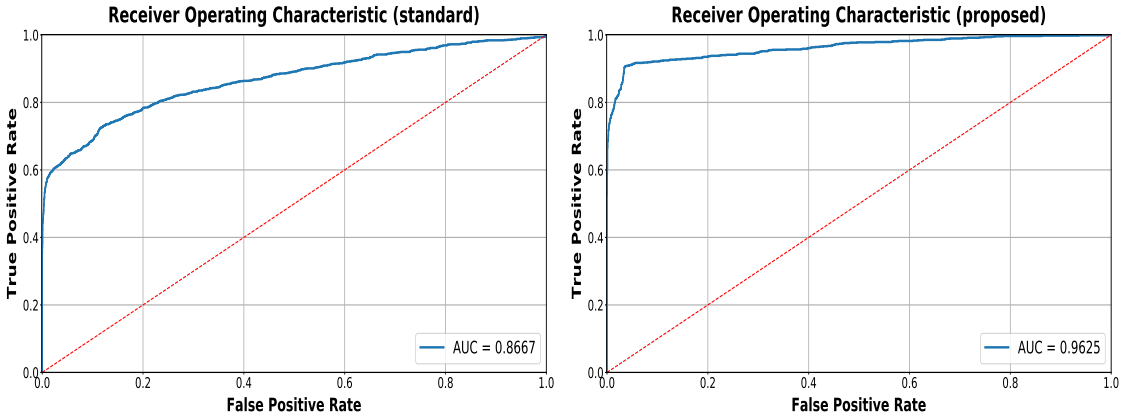
Figure 7.10: Precision and recall for different threshold values.

recall greater than 0.9 when trained on six seconds compacted PPG windows with data augmentation. During the experiments, it was noticed that the larger the data window, the easier it was for the model to detect anomalies since large PPG windows contain more features and enable the model to learn more about the regular shape of PPG signals. The improved results of the proposed approach can be explained by the fact that we can allow the model to learn the patterns and shape of larger windows of PPG data from a summarized version by using the discrete wavelet transform.

Figure 7.11 shows that the proposed approach have a higher area under the ROC curve (AUC) than the standard approach. Note that a model with an AUC higher than 0.5 is better than a random classifier. Selecting a threshold that gives a true positive rate greater than 0.9 and a false positive rate close to 0 is possible when using data augmentation.

In order to define whether or not a PPG signal is valid, we defined the cut point as 0.06, this threshold is based on the last training loss value obtained during the training. If the mean squared error between the reconstructed signal and the input exceeds this threshold then this signal is marked as a meaningless PPG signal. Figure 7.12 shows the reconstruction of one clean PPG signal and two irrelevant signals. The mean squared error when the signal is clean is 0.02, while it exceeds the threshold value for the other two signals. It was noticed that there was a reconstruction error greater than 0.1 in most PPG signals with artifacts.

The trained model was tested on the test set containing a mix of clean and irrelevant PPG windows. The test set is unlabeled, so a visual assessment was needed to verify the model's efficacy. Figure 7.13 displays four different parts taken from the test set. The green zone is where the model predicted the signal as clean and the red zone as artifacts.



(a) (b)

Figure 7.11: Receiver operating characteristic curves for the standard and proposed approaches.

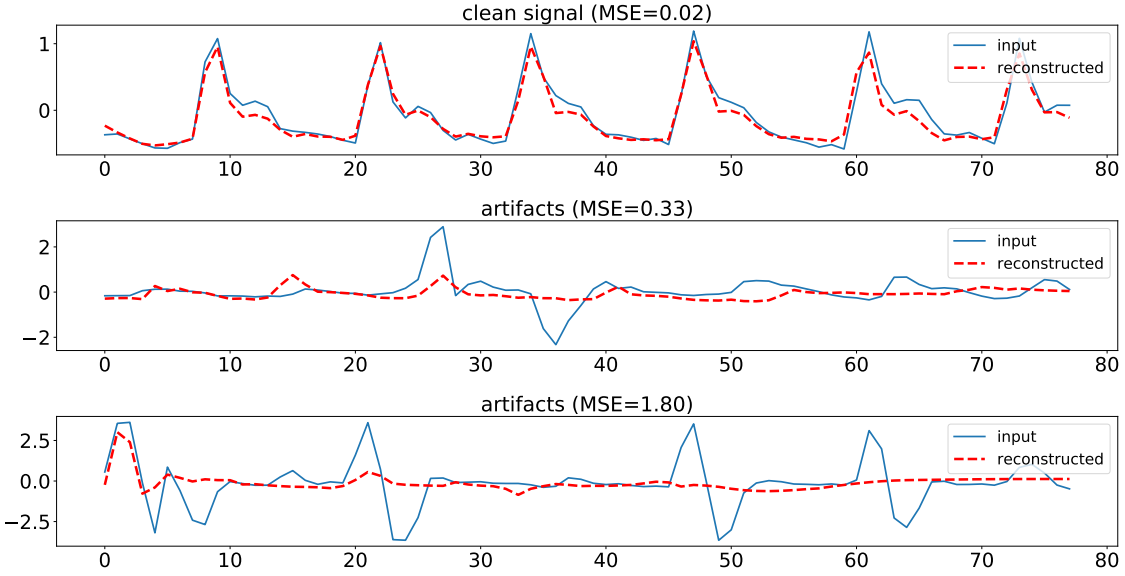
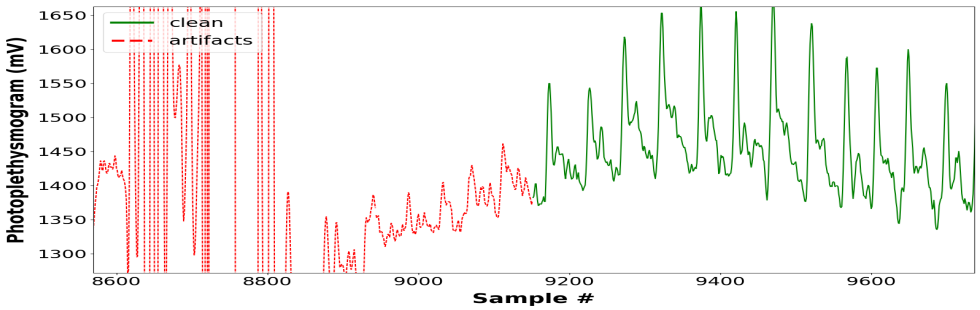
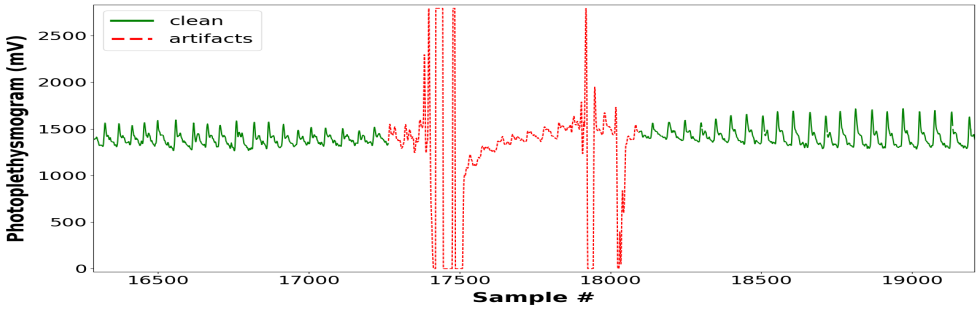


Figure 7.12: Examples of mean squared error obtained when clean PPG signals and artifact signals are reconstructed.

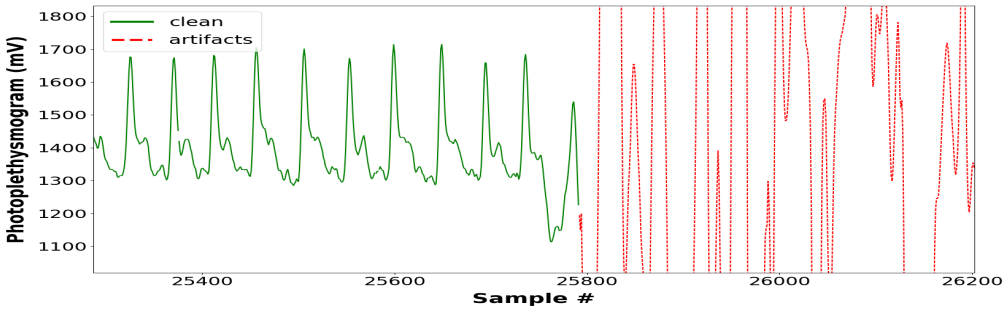
From the figures it can be seen that the model can efficiently predict the irrelevant PPG windows where it is difficult to extract important features. This section, in addition to summarizing data, proposed a data augmentation process for medical time series collected from wearable devices and sensors to increase the amount of training data and make the model more robust to noise.



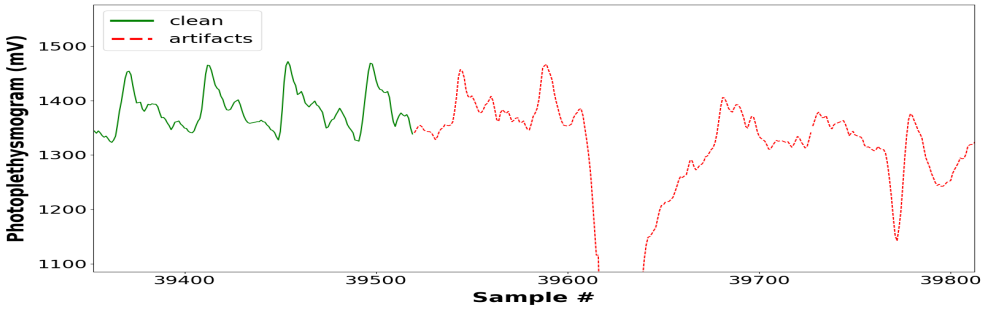
(a)



(b)



(c)



(d)

Figure 7.13: Four windows taken from the test set showing the prediction results obtained from the trained model.

7.6/ CONCLUSION

The contribution in this chapter is twofold: First, a convolutional autoencoder-based deep learning model was introduced to solve the limitations discussed in Chapter 5, namely the distortion of the signals when compressed using SZ with high error-bound value. The proposed model was able to enhance the compressed data after reconstruction, thus fixing the shape of the signal and allowing the features to be extracted more accurately. Second, a deep learning model for automatic motion artifacts detection from photoplethysmography (PPG) was proposed.

Given the advancements in wearables and the capability of most modern smartwatches to capture PPG data, in addition to PPG's importance in stress detection, PPG's analysis and processing is of considerable importance. Before training the model, two major steps were taken that helped to achieve better results. First, a data summarization step was applied to the input data through the discrete wavelet transform to reduce the length of the sequences and avoid vanishing gradients. As a result, summarizing large windows can still allow the model to learn the same patterns while processing less data points, the model can better learn patterns from larger data windows than small ones.

Finally, a CNN-LSTM autoencoder architecture was proposed to detect and discard irrelevant zones in photoplethysmogram signals in order to avoid analyzing and processing meaningless data. The findings of this work show that the proposed approach to data summarization and augmentation helped to improve the performance of the neural network and was able to achieve 90% precision and 95% recall. The proposed method for PPG artifacts filtering can be generalized for various types of medical time series such as electrocardiogram (ECG) data in addition to periodic sensory data.

8

USING DENSENET FOR IOT MULTIVARIATE TIME SERIES CLASSIFICATION

Time series classification task was first discussed in Chapter 6. This chapter examines the use of the DenseNet architecture, originally proposed for computer vision applications, for the classification of multivariate time series. Nowadays, most Internet of Things (IoT) devices collect multiple features and produce multivariate time series. In an IoT application, the mining and classification of the collected data have become crucial tasks. Hybrid LSTM-fully convolutional networks (MLSTM-FCN) provide state-of-the-art classification results on multivariate time series benchmarks. We propose in this chapter a hybrid LSTM-DenseNet model that is able to achieve the performance of the state-of-the-art models and surpass them in many situations, based on the results obtained from various experiments on 15 benchmark datasets. Thus, this work suggests the 1D DenseNet as a potential tool to be considered by machine learning engineers and data scientists for IoT time series classification in healthcare applications.

8.1/ INTRODUCTION

Efficient real-time edge analytics covers numerous functionalities in data mining, such as classification, clustering, outlier detection, time series analysis and association analysis, and has recently been attracting many researchers [205]. Many of the data collected from IoT-healthcare and Industrial IoT (IIOT) applications have been gathered over the course of time, constituting a time series. Time Series Classification (TSC) has been used to solve recognition tasks in different areas, from healthcare to science and industry. These tasks include signature verification, driver guidance, activity recognition, cardiovascular disease detection, and neurological disorders [98, 116, 134]. With the development of the

IoT and 5G, the learning representations and the TSC become more and more relevant.

A time series is a set of ordered real values. Nowadays, many of the IoT devices can collect multiple features, resulting in Multivariate Time Series (MTS). A MTS $T = [t^1, t^2, ..., t^n]$ consists of n univariate time series, where $t^i \in I\!R^L$ and L is the length of each time series t^i. Various time series classification methods were proposed in the literature [140]. The most famous classifiers are the distance-based classifiers such as k-nearest neighbor with Dynamic Time Warping (DTW) distance and dictionary-based classifiers in which a time series is converted into representative words. An example of dictionary-based techniques is the bag-of-SFA-symbols (BOSS). Certain classifiers are shapelet-based, such as the Fast Shapelets (FS) and the Shapelet Transform (ST), and ensemble-based, such as the Collective of Transformation Ensembles (COTE), which is a combination of classifiers in different domains. The aforementioned categories and methods of classification are described in detail in [112, 140].

On the other hand, deep learning has offered significant neural architectures that are designed to address the particular structural features of image and language data. For instance, convolutional neural networks (CNNs) are built to take advantage of two- or three-dimensional image structures, whereas most language models use recurrent neural networks (RNNs) that embrace sequence and memory naturally found in both spoken and written languages. Such recent technologies were also extended to areas where classical machine learning has historically dominated. Time Series Classification (TSC) is one such field.

End-to-end deep learning approach to time series classification has proved to be effective in recent years [201]. This approach consists in learning the hidden features from raw time series using non-linear transformations without the need for extensive feature engineering. Different architectures have been proposed in the literature that are based on Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Multi Layer Perceptron (MLP), and Echo State Network [164, 201].

This chapter proposes a Deep Neural Network (DNN) model that is based on the densely connected convolutional network, namely DenseNet, and the Long Short-Term Memory (LSTM) RNN. Note that the objective of this chapter is not to compare the proposed deep learning model with classifiers in different categories (distance-based, feature-based, ensemble, etc.), but rather to show that DenseNets are candidates which should be taken into account when choosing a deep learning architecture for TSC. Note that when processing time series, DenseNets can be used in a stand-alone way, or combined with LSTMs and the work of this chapter will consider both strategies.

8.2/ BACKGROUND

This section provides a reminder from Time Series Classification (TSC) perspective on CNNs and RNNs, as most of the proposed TSC models are based on convolution blocks and LSTM units.

8.2.1/ CONVOLUTIONAL NEURAL NETWORKS FOR TSC

Normally, CNNs are used to analyze image data. With multidimensional data, such as images and MTS in our case, this kind of network scales well. A CNN is a feed forward network consisting of many layers including convolutional filters with Batch Normalization (BN) [119] and layers for rectification and pooling. A CNN typically adopts one or more fully-connected layers after the last pooling layer in order to convert 2-D feature maps into a 1-D vector to allow for final classification.

CNNs have proven to be effective in solving computer vision issues, including achieving state-of-the-art results on tasks such as image classification and captioning, object localization and more. This is done by working directly on raw data such as raw pixel values instead of on manually derived features extracted from the raw data. In order to solve the addressed problem, the model learns how to automatically extract the useful features from the raw data. This type of learning is referred to as representation learning, and the CNN does this in a way that extracts the features regardless of how they appear in the data. CNNs' ability to learn and derive features automatically from raw input data can be extended to TSC problems. A series of observations can be viewed as an image that can be read and transformed into the most salient elements by a CNN model. In this case, a convolution can be seen as applying and sliding a one-dimensional filter over the time series to perform a non-linear transformation.

8.2.2/ RECURRENT NEURAL NETWORKS FOR TSC

Recurrent neural networks such as the Long Short-Term Memory Network (LSTM) provide the explicit handling of order between the input samples while learning a mapping function from inputs to outputs that CNNs do not provide. Typically, RNNs are used with training samples that have clear interdependencies and relevant representation to hold information about what happened in previous time steps. In other words, these models are providing output by leveraging two input sources, the current and the recent past.

RNNs suffer from the so-called long-term dependencies problem. This means that when attempting to link previous information to a present task they can not reach far previous memory. LSTM networks can be effective for TSC problems because they can overcome

the long-term dependencies issue by incorporating weights and gates [4], and they have the ability to learn long-term associations in a sequence. LSTM networks do not require a pre-specified time window and are able to model complex multivariate sequences accurately. However, even if an LSTM is expected to capture the long-term dependencies better than the RNN, it appears to become forgetful. This issue is tackled by *Bahdanau et al* in [87]. *Bahdanau et al* proposed the attention mechanism to address the long-term dependencies problem in neural-based machine translation of texts. The goal was to assign some of the input words more significance compared to others while the sentence was being translated. The authors have done this simply by building a context vector and taking a weighted sum of the hidden states. The core idea is an attempt to mimic the human brain by selectively concentrating on a few relevant things, while ignoring others. The authors in [207] detail the use of the attention mechanism for TSC.

8.3/ RELATED WORK

Different neural network architectures for multivariate TSC were proposed in the literature. In the following we will present the most effective architectures based on the results obtained on the TSC benchmarks, namely the UCR/UEA archive [113, 140] and the MTS archive [240], in addition to the conclusions derived from detailed surveys [164, 201].

The Fully Convolutional Networks (FCNs) were originally proposed for segmentation tasks [99, 177] and are very effective in extracting features from input data. The FCN used for TSC is formed by stacking three blocks, each consisting of a convolutional layer with filters, preceded by a batch normalization layer and a ReLU activation layer as shown in Figure 8.1-a. Then a global average pooling layer is applied to the features after the first three convolutional blocks, effectively reducing the number of weights. Finally, the softmax layer produces the final output.

The Residual Network (ResNet) is a deep architecture used for TSC. There are many variants. The version presented in Figure 8.1-b consists of 9 convolutional layers followed by global average pooling and softmax layers. This architecture uses shortcut connections between successive convolutional layers for training and adds linear shortcuts to connect a residual block's output to its input thus resulting in an easier training.

The authors in [207] proposed the Multivariate LSTM Fully Convolutional Network (MLSTM-FCN). This proposed architecture has yielded so far the best results on the multivariate UEA archive. The MLSTM-FCN model, as shown in Figure 8.1-c, consists of a fully convolutional block and a LSTM block. The input multivariate time series is transferred into a shuffle layer and subsequently passing it into a LSTM block with attention mechanism followed by dropout. The attention LSTM layer output is concatenated with

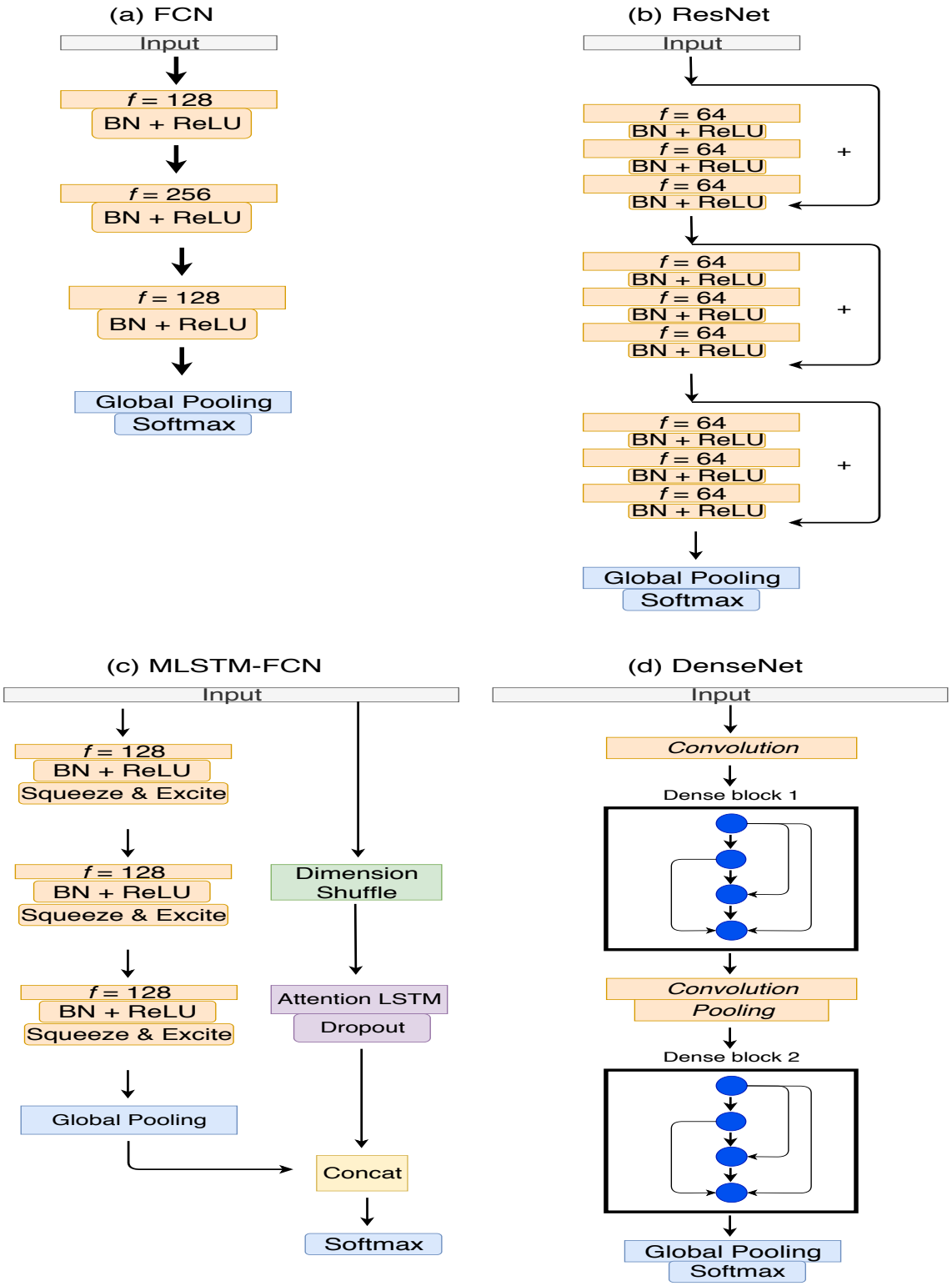


Figure 8.1: The network structure of FCN, ResNet, MLSTM-FCN, and DenseNet.

the global pooling layer output, and the final results are obtained from the softmax layer. An interesting approach used by the authors is the addition of the squeeze and excite block proposed and detailed in [175] after each convolutional block, thus enhancing the

performance of the model.

The Densely Connected Convolutional Networks (DenseNet) was proposed in [150]. The proposed architecture attempts to solve the problem of vanishing gradient by introducing connection from one layer to all its consequent layers in a feed forward manner. The authors in [231] adapted the DenseNet architecture for univariate TSC. The motivation behind using DenseNet for time series is that this architecture facilitates the propagation of features and enables feature reuse, which greatly reduces the number of parameters. However, the main objective of the work in [231] is to study the normalization techniques used with DenseNet and not to compare its performance with the state of the art DNN techniques used for TSC.

The rest of this chapter introduces an adapted version of DenseNet for the processing of multivariate time series (Figure 8.1-d), and a new architecture inspired by the work described in [207] in which the DenseNet is used with LSTM. Notice that the authors in [207] referred to their architecture as MALSTM-FCN, where they use the attention mechanism with LSTM. We will refer to this architecture in this work as MLSTM-FCN.

8.4/ PROPOSED MODELS

This section presents the two DNN models considered in this work. Similarly to [231], the first model is an adapted version of the DenseNet for multivariate time series, and is illustrated in Figure 8.2. It takes a signal input of size (m, n) where m is the number of timesteps and n the number of features. The input goes first through a 1D convolution layer with a kernel size of 3 then through two dense blocks and a global pooling step. Each dense block consists of four convolution steps. Each step applies four operations, namely batch normalization, ReLU activation, squeeze and excite operation inspired from [207] as well as a 1D convolution with kernel size of 3. The number of filters is initialized to 32 and incremented by 16 after each step. It may be noticed that, after each step, the number of parameters in each dense block increases by more than 16. This is due to the concatenation operation, where features of the k^{th} layer are concatenated with features of previous layers and are provided as input to the next layer. Between both dense blocks, a down-sampling layer that reduces the size of feature maps is used. This layer is referred to as the transition layer and it applies batch normalization, ReLU activation, 1D convolution with kernel size 1 and 1D average pooling with pooling size 2 to the input. The last dense block's output is passed through batch normalization, ReLU activation, 1D global average pooling, and softmax.

The second model uses the same DenseNet architecture of the first, and adds more complexity to it. It is illustrated in Figure 6.2-c and inspired from the work proposed

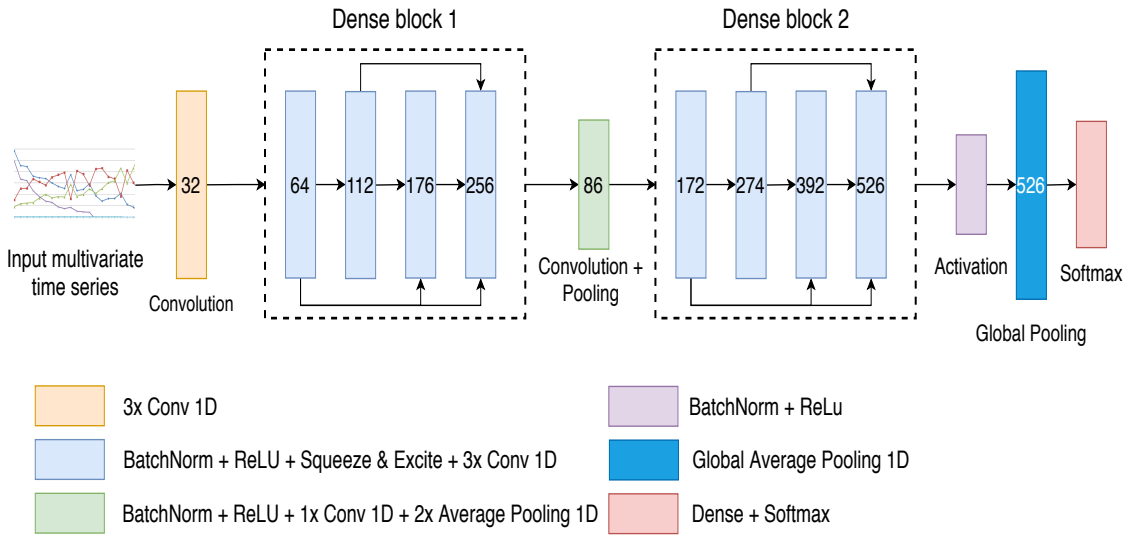


Figure 8.2: The adapted 1D-DenseNet architecture for multivariate time series classification. The numbers in each box denote the number of parameters at each step.

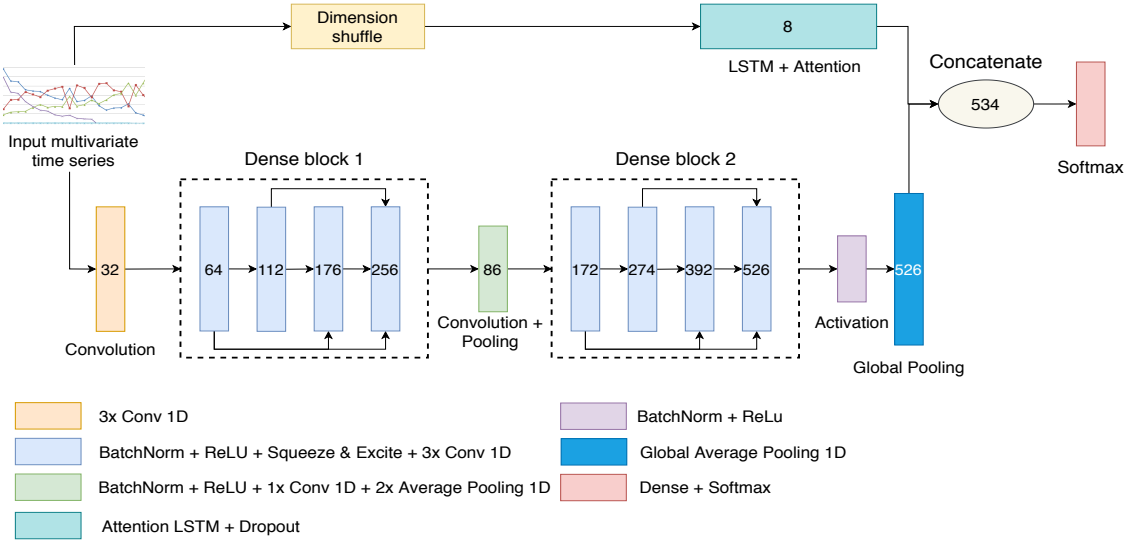


Figure 8.3: The adapted MLSTM-FCN architecture proposed in [207] where the FCN is replaced by the 1D-DenseNet. The numbers in each box denote the number of parameters at each step.

in [207]. As shown in Figure 8.3, a signal input of size (m, n) is fed to the previously described dense architecture, in addition to a LSTM layer with 8 units and an attention mechanism followed by a dropout operation. The number of units has been chosen based on the results achieved in [207]. Note that the temporal dimension of the input is shuffled before being fed to the LSTM layer when the number of timesteps is greater than the number of features. Given that the number of timesteps m is usually greater than the number of features n in a time series, the LSTM will involve n timesteps to handle m features, which is more efficient, when shuffling the dimension and providing the LSTM an input of dimension (n, m). The output of the attention LSTM layer is then concatenated

with the output of the global pooling layer of the DenseNet block and passed through a softmax layer. Both models use the variant of stochastic gradient descent 'Adam' [96] with a learning rate of 0.001 as optimizer[1].

8.5/ EXPERIMENTS

In this work we use the Keras library [114] with the Tensorflow backend [108] to train the aforementioned models. In the experiments of this section, 15 datasets coming from [46, 172, 240] are used. Note that the authors in previous works have made their experiments on more datasets, such as in [201, 207]. The previously proposed models were able to achieve very high accuracy on multiple datasets, ranging from 95% to 100%. Such datasets are omitted in our experiments where very high accuracy is reached by most of the DNN models, since the features in these datasets can be easily learned by a simple classifier and there is nothing to improve. The computations were performed on an NVIDIA Tesla Titan X GPU to train the models on these datasets.

A DNN model's input is a tensor with shape (s, m, n), where s is the number of samples, m is the length of each sample's time series (or number of timesteps), and n is the number of features. The samples of each dataset are already divided into train and test sets. Hyperparameters such as the number of epochs and batch size are specified for each dataset, and the option of standardizing the numerical features to have a mean 0 and unit variance using standard scalar is based on the numerical ranges of the features and the various empirical tests performed. Table 8.1 shows the properties and parameters used for each dataset.

The accuracy metric was used to evaluate the performance of the five DNN models discussed in this section, namely FCN, ResNet, MLSTM-FCN and the two models described in section IV: DenseNet and MLSTM-DenseNet. Important to note that the accuracy value achieved in the experiments by the state-of-the-art models may vary from the one achieved in the original papers, and this is due to the change in the hyperparameters, and the standardization of some datasets in addition to the stochastic nature of deep learning algorithms. Though, we assured that all models are trained and evaluated on each dataset with the same settings and conditions.

Table 8.2 shows the classification accuracy value obtained by implementing the 1D-DenseNet and MLSTM-DenseNet, and regenerating the FCN, ResNet, and MLSTM-FCN models based on their actual implementation on github. The proposed MLSTM-DenseNet model was able to outperform the other models on 8 datasets and provide the highest accuracy on 3 datasets, namely ECG, LP2, and KickvsPunch, alongside the MLSTM-FCN.

[1]The codes and weights of some models are available at https://github.com/josephazar/MLSTM-DenseNet

Dataset	# classes	# features	Timesteps	# samples	Task	Train-test	Batch size	Epochs	Standardize	Source
Uwave	8	3	315	4478	Gesture recognition	20-80	16	2000	yes	[240]
Gesture Phase	5	18	214	396	Gesture recognition	50-50	16	2000	yes	[172]
LP3	4	6	15	47	Robot failure recognition	36-64	8	2000	no	[172]
LP5	5	6	15	164	Robot failure recognition	39-61	16	2000	no	[172]
Ozone	2	72	291	344	Weather classification	50-50	16	2000	no	[172]
ECG	2	2	152	200	ECG classification	50-50	16	2000	yes	[240]
EEG	2	13	117	128	EEG classification	50-50	16	2000	no	[172]
KickvsPunch	2	62	841	26	Action recognition	62-38	4	2000	yes	[240]
Netflow	2	4	997	1337	Action recognition	60-40	128	2000	yes	[240]
Action 3D	20	570	100	567	Action recognition	47-53	16	2000	yes	[46]
LP2	5	6	15	47	Robot failure recognition	36-64	8	2000	no	[172]
HT Sensor	3	11	5396	100	Food classification	50-50	8	2000	no	[172]
Movement AAL	2	4	119	314	Movement classification	50-50	64	2000	yes	[172]
Occupancy	2	5	3758	117	Occupancy classification	35-65	16	2000	yes	[172]
AREM	7	7	480	82	Activity recognition	51-49	16	2000	no	[172]

Table 8.1: Properties of all the datasets used in this chapter and originated from [172, 240, 46]. The batch size and number of epochs used to train the DNN models are shown for each dataset.

The 1D-DenseNet model yielded the best accuracy on the Uwave and Netflow datasets, the ResNet model on the HT Sensor dataset, and the MLSTM-FCN model on the Movement AAL dataset. This means that the performance of the proposed MLSTM-DenseNet model over various datasets is generally higher than that of other state-of-the-art models.

Dataset	FCN	ResNet	DenseNet	MLSTM-DenseNet	MLSTM-FCN
Uwave	93.4	92.6	**93.6**	89.7	89.3
Gesture Phase	45.9	47.9	48.9	**57.0**	55.5
LP3	46.6	43.3	70.0	**83.3**	80.0
LP5	68.0	65.0	68.0	**69.0**	67.0
Ozone	81.5	83.2	81.9	**86.0**	82.0
ECG	90.0	91.0	92.0	**93.0**	**93.0**
EEG	64.0	62.5	62.5	**67.0**	65.6
KickvsPunch	60.0	80.0	**100**	**100**	**100**
Netflow	95.5	95.8	**96.6**	95.1	87.2
Action 3D	55.5	57.5	52.1	**77.1**	75.4
LP2	46.3	30.0	76.6	**83.3**	**83.3**
HT Sensor	84.0	**88.0**	82.0	86.0	82.0
Movement AAL	61.1	59.2	58.6	77.0	**79.0**
Occupancy	71.0	72.3	72.3	**79.0**	77.0
AREM	84.6	89.7	87.18	**94.8**	92.3

Table 8.2: Achieved accuracy by the five DNN models on the 15 multivariate time series datasets. The best performance is shown in bold.

8.6/ CONCLUSION

The proposed MLSTM-DenseNet classification model shows that replacing the Fully Convolutional Network (FCN) with a DenseNet is a potential solution that needs to be considered for the classification of multivariate time series. The benefit of DenseNet is that every layer has direct access to the gradients from the loss function and the original input signal, resulting in an enhanced flow of information and gradients across the network, as well as a regularizing effect that decreases overfitting on tasks with limited training set sizes. Experiments have been conducted on 15 multivariate time series datasets, and the results show that the proposed approach is capable of achieving the performance of the state-of-the-art models and bypassing them in several cases. For some datasets, such as the ones in which the number of timesteps is high, the standalone 1D DenseNet outperformed the hybrid models. Considering the 1D DenseNet as a standalone model or as a replacement for FCN in the MLSTM-FCN for the classification of multivariate time series can thus help to improve classification performance.

For future work, we intend to perform further experiments and compare the models with regard to other metrics than the accuracy such as using the mean per-class error, training and prediction time and the complexity of the models. Another important task to be considered in the future is to present an interpretable feedback highlighting the reason for a certain decision taken by the classifier using Class Activation Map as discussed in [201]. Moreover, we intend to better understand why the proposed model does not match the FCN and ResNet performance on the Uwave dataset.

IV

Conclusion & Perspectives

9

CONCLUSION & PERSPECTIVES

9.1/ CONCLUSION

In this thesis, data collection as well as data processing and analysis approaches for wireless body sensor networks and IoT-healthcare have been proposed in order to ensure efficient health monitoring. This dissertation is composed of two parts: the first part covers the scientific background of Iot-healthcare, whereas the second one presents the contributions that have been made in this thesis.

The first part began by presenting some of the Iot-healthcare applications, such as mobile health, ambient assisted living and elderly monitoring, smart medical implants, and detection of stress and mental overload. Then, the data collection, data integration, big data, unobtrusiveness, interoperability, and privacy issues that need to be addressed in such applications are discussed.

Second, the wireless body sensor network and the consumer health wearables technologies are discussed. The major mechanisms of energy-efficient communication, sensing, and processing were presented. Given that the first major goal of this thesis is energy-efficiency with data compression, various practical approaches to IoT data compression have been presented.

Third, the use of artificial intelligence in IoT has been introduced. The ability of machine learning techniques to analyze and extract knowledge automatically from the data being collected has been shown, and medical decision-making frameworks based on machine learning methods have been identified. In addition, Deep learning approaches and their application to various areas and applications in health informatics were discussed.

The second part of this dissertation presented the contributions. The first research done focused on the data collection process of an IoT-healthcare application, specifically the problem of energy efficiency. This work proposed an energy-efficient data reduction scheme based on an error-bound lossy compressor, namely Squeeze (SZ), for IoT-Edge applications in healthcare. A lightweight version of the compressor have been imple-

mented on real IoT hardware and a Polar M600 wearable. The analysis of the trade-off between compression ratio and classification performance has received a great deal of attention in this study. To do so, the use case of driving behavior monitoring in intelligent vehicle systems where vital signs data are collected from the driver using a Wireless Body Sensor Network (WBSN) and wearable devices and sent to an edge node for stress level detection was considered. In addition, a new compression approach has been proposed which combines the discrete wavelet transform lifting scheme with SZ, and which can effectively manage univariate and multivariate time series. The effect of different data compression techniques on the classification performance by state-of-the-art deep learning models on different time series datasets has been studied. The proposed solution proposed in this thesis could efficiently reduce the amount of data to be transmitted, energy consumption, and the time required to transfer data from the IoT to the edge. As a result, the data reduction approach proposed in this thesis can be considered for a wide range of IoT applications requiring an easy-to-implement algorithm that reduces the processing/transmission time and optimize the lifetime of the devices.

The second research that was done focused on the edge-level data analysis and processing task, particularly the use of deep learning techniques to solve numerous challenges. Three challenges were considered in this thesis: time series reconstruction and enhancement after data compression, photoplethysmogram (PPG) time series filtering and removal of corrupted zones, and time series classification task.

The usage of a lossy data compression technique with high compression ratio results in the distortion of the signal. When it comes to medical and physiological data, the quality of the reconstructed data after compression is of high importance. A highly compressed signal can result in an erroneous extraction of features after reconstruction. To solve this issue, a convolutional autoencoder-based deep learning model was introduced. The proposed technique was inspired from the computer vision field, and it was originally proposed for image denoising. The model was adapted to work on time series data and the results showed that it was able to enhance the compressed data after reconstruction, thus fixing the shape of the signal and the allowing the features to be extracted more accurately.

In this thesis, special attention was given to the application of stress detection. Today most commercial smartwatch will collect PPG data. Given the amount of features that can be derived from this type of data, PPG data will open up various possible medical applications. One of the significant applications that can benefit from PPG is stress detection. The biggest challenge facing these applications is the presence of corrupted zone in the signal due to motion or incorrect sensor positioning. A CNN-LSTM autoencoder architecture was proposed to detect and discard irrelevant zones in PPG signals to prevent the analysis and processing of meaningless data. The results obtained were

satisfactory and and the proposed model can be expanded for different types of medical time series such as electrocardiogram (ECG) data in addition to periodic sensory data.

Finally, the time series classification task has been addressed given its significance in physiological time series analysis and stress detection. Various recent time series classification models based on a deep neural network were presented in the chapters of the second part of the dissertation. In the final contribution, a deep learning model was proposed for the time series classification task based on the DenseNet architecture. The proposed MLSTM-DenseNet classification model combines the LSTM architecture with the DenseNet architecture to achieve better results on multivariate time series benchmark datasets. The results show that the proposed approach is able of achieving and, in several cases, bypassing the outcomes of the state-of-the-art models.

We believe that the solutions proposed in this thesis for various challenges in the phases of data collection and analysis in an IoT-healthcare application can serve as additional and efficient solution choices or tools to address the above-mentioned challenges in healthcare.

9.2/ PERSPECTIVES

There are many perspectives to consider in a environment that grows constantly and has immense potential to impact the lives of people. The key focus of this thesis was to suggest solutions for IoT-healthcare applications where the data obtained are physiological time series. Multiple different approaches should be considered for future research such as considering the Generative Adversarial Networks for data augmentation, and further exploiting deep learning on the mobile and edge.

Generative Adversarial Networks (GANs) have demonstrated significant success as a framework for training models to produce realistic data. A GAN is informally defined as an iterative game played between a discriminator and a generator; while the objective of the discriminator is to reduce its loss by learning to recognize real data as real and fake data as fake, the aim of the generator is to reduce its loss by learning to fool the discriminator by converting random noise into fake data.

As already stated, the work of this thesis generally focused on time series data and particularly on physiological time series. A common problem with the considered use case of stress recognition and other medical use cases is class imbalance. This means that samples of the minority class (stress or other health-related problems) are very little in comparison with samples of the majority class (regular or stable condition). The result of such an issue biases the classification algorithms and allows for low generalization. Many traditional approaches have been used to solve the problem of class imbalances, but very

few deal with time series generation. GANs have the potential to increase data and to generate realistic medical synthetic time series. Such frameworks model the processes that generate data and can help represent the first step towards the development of a new approach to predictive systems and benefit the medical community for use in training simulators.

The work done in this thesis targeted applications where physiological data is gathered from IoT devices and sent for processing to an intermediate node (sink, coordinator). As smaller devices are packed with more computational power, Artificial Intelligence (AI) has become more mobile than it was before. With the introduction of AI, mobile and IoT devices have now been transformed into smart devices. Edge deep learning has recently become an interesting research field, as AI-powered device capabilities are expected to grow day by day. As addressed in this dissertation, when it comes to medical applications, the question to be asked is to what degree do the deployment of light deep learning models on the devices to perform the processing tasks in place instead of transmitting the data to the sink is more efficient? Is it more energy efficient, or is the computation energy consumption of these models high? What kinds of medical applications require that the data be processed directly on the device and do not require its transmission and storage?

Personalization means modifying a system to suit the preferences of a particular individual, sometimes in relation to clusters of individuals. The state of the art research has shown, in the example of stress detection, that individual models are better suited to such application than generalized models. The deployment and study of deep learning on the edge becomes more important in such cases. Mobile and wearable device AI algorithms leverage the user-specific data available, such as the collected PPG signal, location, behavior patterns, etc. Edge deep learning is a field worth exploring and opens up potential solutions for IoT-healthcare applications.

Finally the work done in this thesis opens up limitless perspectives in various research lines. One of the future works is to study applications that collect heterogeneous data and not just time series, and to develop deep learning models that can incorporate and learn from a variety of data types. Additionally, lightweight encryption algorithms can be implemented at node level to ensure secure data transmission and data anonymization should be taken into account in medical applications as well.

PUBLICATIONS

JOURNAL PAPERS

- Joseph Azar, Abdallah Makhoul, Raphael Couturier, and Jacques Demerjian. *"Robust IoT Time Series Classification with Data Compression and Deep Learning"*. In **Neurocomputing**, 398, 222-234 (2020).

- Joseph Azar, Abdallah Makhoul, Mahmoud Barhamgi, and Raphael Couturier. *"An energy efficient IoT data compression approach for edge machine learning"*. In **Future Generation Computer Systems**, 96, 168-175 (2019).

CONFERENCE PAPERS

- Joseph Azar, Abdallah Makhoul, and Raphael Couturier. *"Using DenseNet for IoT multivariate time series classification"*. In **IEEE Symposium on Computers and Communications 2020**, Accepted in April 2020.

- Joseph Azar, Carol Habib, Rony Darazi, Abdallah Makhoul and Jacques Demerjian. *"Using Adaptive Sampling and DWT Lifting Scheme for Efficient Data Reduction in Wireless Body Sensor Networks"*. In **2018 IEEE 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)**.

- Joseph Azar, Rony Darazi, Carol Habib, Abdallah Makhoul and Jacques Demerjian. *"Using DWT Lifting Scheme for Lossless Data Compression in Wireless Body Sensor Networks"*. In **2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)**, pages: 1465-1470, June 2018.

- Joseph Azar, Abdallah Makhoul, Rony Darazi, Jacques Demerjian, and Raphael Couturier. *"On the performance of resource-aware compression techniques for vital signs data in wireless body sensor networks"*. In **2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)**, 1-6.

SUBMITTED PAPERS

- Joseph Azar, Gaby Bou Tayeh, Abdallah Makhoul, and Raphael Couturier. *"Efficient IoT Data Compression and Reconstruction Using SZ and Autoencoder"*. In **IEEE Transactions on Systems, Man and Cybernetics: Systems**. Submitted in March 2020.
- Joseph Azar, Abdallah Makhoul, Raphael Couturier, and Jacques Demerjian. *"Deep Recurrent Neural Network-based Autoencoder for Photoplethysmogram Artifacts Filtering"*. In **Computers and Electrical Engineering**. Submitted in December 2019.

BIBLIOGRAPHY

[1] **Towards a definition of the internet of things (iot)**. https://iot.ieee.org/definition.html. Accessed: 2020-01-13.

[2] WILLIAMS, R. J., AND ZIPSER, D. **A learning algorithm for continually running fully recurrent neural networks**. *Neural computation 1*, 2 (1989), 270–280.

[3] DEUTSCH, P., AND GAILLY, J.-L. **Zlib compressed data format specification version 3.3**. Tech. rep., RFC Editor, United States, 1996.

[4] HOCHREITER, S., AND SCHMIDHUBER, J. **Long short-term memory**. *Neural Computation 9*, 8 (1997), 1735–1780.

[5] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. **Gradient-based learning applied to document recognition**. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[6] SWELDENS, W. **The lifting scheme: A construction of second generation wavelets**. *SIAM Journal on Mathematical Analysis 29* (1998), 511–546.

[7] BOROTSCHNIG, H., PALETTA, L., AND PINZ, A. **A comparison of probabilistic, possibilistic and evidence theoretic fusion schemes for active object recognition**. *Computing 62*, 4 (1999), 293–319.

[8] BENABID, A.-L., KRACK, P., BENAZZOUZ, A., LIMOUSIN, P., KOUDSIE, A., AND POLLAK, P. **Deep brain stimulation of the subthalamic nucleus for parkinson's disease: methodologic aspects and clinical criteria.** *Neurology 55*, 12 Suppl 6 (2000), S40–4.

[9] ET AL., A. L. G. **Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals**. *Circulation* (2000).

[10] PARKES, J., BRYANT, J., AND MILNE, R. **Implantable cardioverter defibrillators: arrhythmias. a rapid and systematic review.** *Health Technology Assessment (Winchester, England) 4*, 26 (2000), 1–69.

[11] CHARBADES, S., KAHANE, P., MINOTTI, L., KOUDSIE, A., HIRSH, E., AND BENABID, A.-L. **Deep brain stimulation in epilepsy with particular reference to the subthalamic nucleus**. *Epileptic disorders 4* (2002), S83–S93.

[12] TURK, D. C. **Clinical effectiveness and cost-effectiveness of treatments for patients with chronic pain**. *The Clinical journal of pain 18*, 6 (2002), 355–365.

[13] TZANETAKIS, G., AND COOK, P. **Musical genre classification of audio signals**. *IEEE Transactions on Speech and Audio Processing 10*, 5 (July 2002), 293–302.

[14] SUNGMEE PARK, AND JAYARAMAN, S. **Enhancing the quality of life through wearable technology**. *IEEE Engineering in Medicine and Biology Magazine 22*, 3 (May 2003), 41–48.

[15] C, E., ROMBERG, J., AND TAO, T. **Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information**, 2004.

[16] DELIGIANNAKIS, A., KOTIDIS, Y., AND ROUSSOPOULOS, N. **Compressing historical information in sensor networks**. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data* (New York, NY, USA, 2004), SIGMOD '04, ACM, pp. 527–538.

[17] SHEN, D., CHEN, Z., YANG, Q., ZENG, H.-J., ZHANG, B., LU, Y., AND MA, W.-Y. **Web-page classification through summarization**. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2004), SIGIR '04, ACM, pp. 242–249.

[18] TROHMAN, R. G., KIM, M. H., AND PINSKI, S. L. **Cardiac pacing: the state of the art**. *The Lancet 364*, 9446 (2004), 1701–1719.

[19] ARMSTRONG, T. **Wake-up based power management in multi-hop wireless networks**. *Term Survey Paper, University of Toronto 238* (2005).

[20] HEALEY, J. A., AND PICARD, R. W. **Detecting stress during real-world driving tasks using physiological sensors**. *IEEE Transactions on Intelligent Transportation Systems 6*, 2 (June 2005), 156–166.

[21] SHANKLE, W. R., ROMNEY, A. K., HARA, J., FORTIER, D., DICK, M. B., CHEN, J. M., CHAN, T., AND SUN, X. **Methods to improve the detection of mild cognitive impairment**. *Proceedings of the National Academy of Sciences 102*, 13 (2005), 4919–4924.

[22] SUBASI, A. **Epileptic seizure detection using dynamic wavelet network**. *Expert Systems with Applications 29*, 2 (2005), 343 – 355.

[23] HINTON, G. E., OSINDERO, S., AND TEH, Y.-W. **A fast learning algorithm for deep belief nets**. *Neural computation 18*, 7 (2006), 1527–1554.

[24] HINTON, G. E., AND SALAKHUTDINOV, R. R. **Reducing the dimensionality of data with neural networks**. *science 313*, 5786 (2006), 504–507.

[25] KIM, B. S., AND YOO, S. K. **Motion artifact reduction in photoplethysmography using independent component analysis**. *IEEE Transactions on Biomedical Engineering 53*, 3 (March 2006), 566–568.

[26] LINDSTROM, P., AND ISENBURG, M. **Fast and efficient compression of floating-point data**. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (Sep. 2006), 1245–1250.

[27] SADLER, C. M., AND MARTONOSI, M. **Data compression algorithms for energy-constrained devices in delay tolerant networks**. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems* (New York, NY, USA, 2006), SenSys '06, ACM, pp. 265–278.

[28] SCHLÖGL, A. **Gdf-a general dataformat for biosignals**. *arXiv preprint cs/0608052* (2006).

[29] SUBASI, A., YILMAZ, M., AND OZCALIK, H. R. **Classification of emg signals using wavelet neural network**. *Journal of neuroscience methods 156*, 1-2 (2006), 360–367.

[30] YANG, G. Z. **Body Sensor Networks**. Springer, 2006.

[31] ZHANG, T., WANG, J., XU, L., AND LIU, P. **Fall detection by wearable sensor and one-class svm algorithm**. In *Intelligent computing in signal processing and pattern recognition*. Springer, 2006, pp. 858–863.

[32] FLOREA, M. C., JOUSSELME, A.-L., AND BOSSÉ, É. **Fusion of imperfect information in the unified framework of random sets theory: application to target identification**. Tech. rep., DEFENCE RESEARCH AND DEVELOPMENT CANADA VALCARTIER (QUEBEC), 2007.

[33] POLAT, K., AND GÜNEŞ, S. **Classification of epileptiform eeg using a hybrid system based on decision tree classifier and fast fourier transform**. *Applied Mathematics and Computation 187*, 2 (2007), 1017–1026.

[34] OBAYYA, M., AND ABOU-CHADI, F. **Data fusion for heart diseases classification using multi-layer feed forward neural network**. In *2008 International Conference on Computer Engineering & Systems* (2008), IEEE, pp. 67–70.

[35] ALIPPI, C., ANASTASI, G., DI FRANCESCO, M., AND ROVERI, M. **Energy management in wireless sensor networks with energy-hungry sensors**. *IEEE Instrumentation & Measurement Magazine 12*, 2 (2009), 16–23.

[36] ANASTASI, G., CONTI, M., DI FRANCESCO, M., AND PASSARELLA, A. **Energy conservation in wireless sensor networks: A survey**. *Ad hoc networks 7*, 3 (2009), 537–568.

[37] ARRABI, S., AND LACH, J. **Adaptive lossless compression in wireless body sensor networks**. In *Proceedings of the Fourth International Conference on Body Area Networks* (ICST, Brussels, Belgium, Belgium, 2009), BodyNets '09, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 19:1–19:8.

[38] COLEY, E., FARHADI, R., LEWIS, S., WHITTLE, I., COLEY, E., FARHADI, R., LEWIS, S., AND WHITTLE, I. **The incidence of seizures following deep brain stimulating electrode implantation for movement disorders, pain and psychiatric conditions**. *British journal of neurosurgery 23*, 2 (2009), 179–183.

[39] EAGLE, N., PENTLAND, A. S., AND LAZER, D. **Inferring friendship network structure by using mobile phone data**. *Proceedings of the national academy of sciences 106*, 36 (2009), 15274–15278.

[40] POON, C., AND IRWIN, M. **Anaesthesia for deep brain stimulation and in patients with implanted neurostimulator devices**. *British journal of anaesthesia 103*, 2 (2009), 152–165.

[41] SALAKHUTDINOV, R., AND HINTON, G. **Deep boltzmann machines**. In *Artificial intelligence and statistics* (2009), pp. 448–455.

[42] CENNINI, G., ARGUEL, J., AKŞIT, K., AND VAN LEEST, A. **Heart rate monitoring via remote photoplethysmography with motion artifacts reduction**. *Opt. Express 18*, 5 (Mar 2010), 4867–4875.

[43] CONCHA, O. P., XU, R. Y. D., AND PICCARDI, M. **Compressive sensing of time series for human action recognition**. In *2010 International Conference on Digital Image Computing: Techniques and Applications* (Dec 2010), pp. 454–461.

[44] HONG, H. G., AND HE, X. **Prediction of functional status for the elderly based on a new ordinal regression model**. *Journal of the American Statistical Association 105*, 491 (2010), 930–941.

[45] LEE, B., HAN, J., BAEK, H. J., SHIN, J. H., PARK, K. S., AND YI, W. J. **Improved elimination of motion artifacts from a photoplethysmographic signal using a kalman smoother with simultaneous accelerometry**. *Physiological Measurement 31*, 12 (oct 2010), 1585–1603.

[46] LI, W., ZHANG, Z., AND LIU, Z. **Action recognition based on a bag of 3d points**. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops* (June 2010), pp. 9–14.

[47] MIETTINEN, A. P., AND NURMINEN, J. K. **Energy efficiency of mobile clients in cloud computing**. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in*

Cloud Computing (Berkeley, CA, USA, 2010), HotCloud'10, USENIX Association, pp. 4–4.

[48] MURUGAN, S., AND RADHAKRISHNAN, S. **Automated ischemic beat classification using genetic algorithm based principal component analysis**. *International Journal of Healthcare Technology and Management 11*, 3 (2010), 151–162.

[49] NAIR, V., AND HINTON, G. E. **Rectified linear units improve restricted boltzmann machines**. In *Proceedings of the 27th International Conference on International Conference on Machine Learning* (USA, 2010), ICML'10, Omnipress, pp. 807–814.

[50] PANTELOPOULOS, A., AND BOURBAKIS, N. G. **A survey on wearable sensor-based systems for health monitoring and prognosis**. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 40*, 1 (Jan 2010), 1–12.

[51] RAGHURAM, M., VENU MADHAV, K., HARI KRISHNA, E., AND ASHOKA REDDY, K. **Evaluation of wavelets for reduction of motion artifacts in photoplethysmographic signals**. In *10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)* (May 2010), pp. 460–463.

[52] VAN MECHELEN, I., AND SMILDE, A. K. **A generic linked-mode decomposition model for data fusion**. *Chemometrics and Intelligent Laboratory Systems 104*, 1 (2010), 83–94.

[53] YILMAZ, T., FOSTER, R., AND HAO, Y. **Detecting vital signs with wearable wireless sensors**. *Sensors 10*, 12 (2010), 10837–10862.

[54] BAJAJ, V., AND PACHORI, R. B. **Classification of seizure and nonseizure eeg signals using empirical mode decomposition**. *IEEE Transactions on Information Technology in Biomedicine 16*, 6 (2011), 1135–1142.

[55] CHAOVALIT, P., GANGOPADHYAY, A., KARABATIS, G., AND CHEN, Z. **Discrete wavelet transform-based time series analysis and mining**. *ACM Comput. Surv. 43*, 2 (Feb. 2011), 6:1–6:37.

[56] LEE, S., LE, H. X., NGO, H. Q., KIM, H. I., HAN, M., LEE, Y.-K., AND OTHERS. **Semi-markov conditional random fields for accelerometer-based activity recognition**. *Applied Intelligence 35*, 2 (2011), 226–241.

[57] MONTE-MORENO, E. **Non-invasive estimate of blood glucose and blood pressure from a photoplethysmograph by means of machine learning techniques**. *Artificial intelligence in medicine 53*, 2 (2011), 127–138.

[58] SCHWARTZ LONGACRE, L., KLONER, R. A., ARAI, A. E., BAINES, C. P., BOLLI, R., BRAUNWALD, E., DOWNEY, J., GIBBONS, R. J., GOTTLIEB, R. A., HEUSCH, G., AND OTHERS. **New horizons in cardioprotection: recommendations from the 2010 national heart, lung, and blood institute workshop**. *Circulation 124*, 10 (2011), 1172–1179.

[59] VICENTE, J., LAGUNA, P., BARTRA, A., AND BAILÓN, R. **Detection of driver's drowsiness by means of hrv analysis**. *Computing in Cardiology 38* (01 2011).

[60] WANG, H., CHOI, H.-S., AGOULMINE, N., DEEN, M. J., AND HONG, J. W.-K. **Information-based energy efficient sensor selection in wireless body area networks**. In *2011 IEEE International Conference on Communications (ICC)* (2011), IEEE, pp. 1–6.

[61] ALBERT, M. V., KORDING, K., HERRMANN, M., AND JAYARAMAN, A. **Fall classification by machine learning using mobile phones**. *PloS one 7*, 5 (2012).

[62] ELGENDI, M. **On the analysis of fingertip photoplethysmogram signals**. *Current Cardiology Reviews 8*, 1 (2012), 14–25.

[63] FAUST, O., ACHARYA, U. R., MA, J., MIN, L. C., AND TAMURA, T. **Compressed sampling for heart rate monitoring**. *Computer methods and programs in biomedicine 108*, 3 (2012), 1191–1198.

[64] FELDMAN, B., MARTIN, E. M., AND SKOTNES, T. **Big data in healthcare hype and hope**. *Dr. Bonnie 360* (2012), 122–125.

[65] KARAN, O., BAYRAKTAR, C., GÜMÜŞKAYA, H., AND KARLIK, B. **Diagnosing diabetes using neural networks on small mobile devices**. *Expert Systems with Applications 39*, 1 (2012), 54–60.

[66] LARANJO, I., MACEDO, J., AND SANTOS, A. **Internet of things for medication control: Service implementation and testing**. *Procedia Technology 5* (2012), 777 – 786. 4th Conference of ENTERprise Information Systems – aligning technology, organizations and people (CENTERIS 2012).

[67] NILSEN, W., KUMAR, S., SHAR, A., VAROQUIERS, C., WILEY, T., RILEY, W. T., PAVEL, M., AND ATIENZA, A. A. **Advancing the science of mhealth**. *Journal of Health Communication 17*, sup1 (2012), 5–10.

[68] PETERSEN, J., LARIMER, N., KAYE, J. A., PAVEL, M., AND HAYES, T. L. **Svm to detect the presence of visitors in a smart home environment**. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2012), IEEE, pp. 5850–5853.

[69] RAM, M. R., MADHAV, K. V., KRISHNA, E. H., KOMALLA, N. R., AND REDDY, K. A. **A novel approach for motion artifact reduction in ppg signals based on as-lms adaptive filter**. *IEEE Transactions on Instrumentation and Measurement 61*, 5 (May 2012), 1445–1457.

[70] SRISOOKSAI, T., KEAMARUNGSI, K., LAMSRICHAN, P., AND ARAKI, K. **Practical data compression in wireless sensor networks: A survey**. *Journal of network and computer applications 35*, 1 (2012), 37–59.

[71] AMER, M., GOLDSTEIN, M., AND ABDENNADHER, S. **Enhancing one-class support vector machines for unsupervised anomaly detection**. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description* (2013), pp. 8–15.

[72] ANGUITA, D., GHIO, A., ONETO, L., PARRA, X., AND REYES-ORTIZ, J. L. **A public domain dataset for human activity recognition using smartphones**. In *ESANN* (2013).

[73] FAKOOR, R., LADHAK, F., NAZI, A., AND HUBER, M. **Using deep learning to enhance cancer diagnosis and classification**. In *Proceedings of the international conference on machine learning* (2013), vol. 28, ACM New York, USA.

[74] HARDAKER, P. A., PASSOW, B. N., AND ELIZONDO, D. **State detection from electromyographic signals towards the control of prosthetic limbs**. In *2013 13th UK Workshop on Computational Intelligence (UKCI)* (2013), IEEE, pp. 120–127.

[75] INCEL, O. D., KOSE, M., AND ERSOY, C. **A review and taxonomy of activity recognition on mobile phones**. *BioNanoScience 3*, 2 (2013), 145–171.

[76] KHALEGHI, B., KHAMIS, A., KARRAY, F. O., AND RAZAVI, S. N. **Multisensor data fusion: A review of the state-of-the-art**. *Information fusion 14*, 1 (2013), 28–44.

[77] LAI, X., LIU, Q., WEI, X., WANG, W., ZHOU, G., AND HAN, G. **A survey of body sensor networks**. *Sensors 13* (2013), 5406 – 5447.

[78] LAIYMANI, D., AND MAKHOUL, A. **Adaptive data collection approach for periodic sensor networks**. In *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)* (2013), IEEE.

[79] LI, S., DA XU, L., AND WANG, X. **A continuous biomedical signal acquisition system based on compressed sensing in body sensor networks**. *IEEE transactions on industrial informatics 9*, 3 (2013), 1764–1771.

[80] LIM, L., MISRA, A., AND MO, T. **Adaptive data acquisition strategies for energy-efficient, smartphone-based, continuous processing of sensor streams**. *Distributed and Parallel Databases 31*, 2 (2013), 321–351.

[81] MAHN, T. G. **Wireless medical technologies: Navigating government regulation in the new medical age**. *Fish's Regulatory & Government Affairs Group* (2013).

[82] MANYIKA, J., CHUI, M., BUGHIN, J., DOBBS, R., BISSON, P., AND MARRS, A. **Disruptive technologies: Advances that will transform life**. *Business, and the Global Economy 180* (2013).

[83] RAZZAQUE, M. A., BLEAKLEY, C., AND DOBSON, S. **Compression in wireless sensor networks: A survey and comparative evaluation**. *ACM Transactions on Sensor Networks (TOSN) 10*, 1 (2013), 1–44.

[84] STEEL, E., AND DEMBOSKY, A. **Health apps run into privacy snags**. *Financial Times* (2013).

[85] STEINHUBL, S. R., MUSE, E. D., AND TOPOL, E. J. **Can Mobile Health Technologies Transform Health Care?** *JAMA 310*, 22 (12 2013), 2395–2396.

[86] ZOIS, D.-S., LEVORATO, M., AND MITRA, U. **Energy-efficient, heterogeneous sensor selection for physical activity detection in wireless body area networks**. *IEEE Transactions on signal processing 61*, 7 (2013), 1581–1594.

[87] BAHDANAU, D., CHO, K., AND BENGIO, Y. **Neural machine translation by jointly learning to align and translate**. *CoRR abs/1409.0473* (2014).

[88] BANGASH, J., ABDULLAH, H., ANISI, H., AND KHAN, A. **A survey of routing protocols in wireless body sensor networks**. *Sensors (Basel, Switzerland) 14* (01 2014), 1322–57.

[89] BANOS, O., GALVEZ, J.-M., DAMAS, M., POMARES, H., AND ROJAS, I. **Window size impact in human activity recognition**. *Sensors 14*, 4 (2014), 6474–6499.

[90] CAVALLARI, R., MARTELLI, F., ROSINI, R., BURATTI, C., AND VERDONE, R. **A survey on wireless body area networks: Technologies and design challenges**. *IEEE Communications Surveys & Tutorials 16*, 3 (2014), 1635–1657.

[91] CHEN, Z., CHEN, Y., HU, L., WANG, S., JIANG, X., MA, X., LANE, N. D., AND CAMPBELL, A. T. **Contextsense: unobtrusive discovery of incremental social context using dynamic bluetooth data**. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication* (2014), pp. 23–26.

[92] COHEN, M. X. **Fundamentals of Time-Frequency Analyses in Matlab/Octave**. sinc(x) Press, 2014.

[93] FRAGKIADAKIS, A., CHARALAMPIDIS, P., AND TRAGOS, E. **Adaptive compressive sensing for energy efficient smart objects in iot applications**. In *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems (VITAE)* (May 2014), pp. 1–5.

[94] GHASEMZADEH, H., AMINI, N., SAEEDI, R., AND SARRAFZADEH, M. **Power-aware computing in wearable sensor networks: An optimal feature selection**. *IEEE Transactions on Mobile Computing 14*, 4 (2014), 800–812.

[95] HABER, D., THOMIK, A. A., AND FAISAL, A. A. **Unsupervised time series segmentation for high-dimensional body sensor network data streams**. In *2014 11th International Conference on Wearable and Implantable Body Sensor Networks* (2014), IEEE, pp. 121–126.

[96] KINGMA, D. P., AND BA, J. **Adam: A method for stochastic optimization**. *CoRR abs/1412.6980* (2014).

[97] KOLDIJK, S., SAPPELLI, M., VERBERNE, S., NEERINCX, M. A., AND KRAAIJ, W. **The swell knowledge work dataset for stress and user modeling research**. In *Proceedings of the 16th international conference on multimodal interaction* (2014), pp. 291–298.

[98] KURBALIJA, V., RADOVANOVIĆ, M., IVANOVIĆ, M., SCHMIDT, D., VON TRZEBIATOWSKI, G. L., BURKHARD, H.-D., AND HINRICHS, C. **Time-series analysis in the medical domain: A study of tacrolimus administration and influence on kidney graft function**. *Computers in Biology and Medicine 50* (2014), 19 – 31.

[99] LONG, J., SHELHAMER, E., AND DARRELL, T. **Fully convolutional networks for semantic segmentation**. *CoRR abs/1411.4038* (2014).

[100] MACIEL, C., DE SOUZA, P. C., VITERBO, J., MENDES, F. F., AND EL FALLAG-SEGHROUCHNI, A. **A multi-agent architecture to support ubiquitous applications in smart environments**. In *Agent Technology for Intelligent Mobile Services and Smart Societies*. Springer, 2014, pp. 106–116.

[101] MANJUSHA, K., SANKARANARAYANAN, K., AND SEENA, P. **Prediction of different dermatological conditions using naive bayesian classification**. *International Journal of Advanced Research in Computer Science and Software Engineering 4*, 1 (2014).

[102] PERERA, C., JAYARAMAN, P. P., ZASLAVSKY, A., CHRISTEN, P., AND GEORGAKOPOULOS, D. **Mosden: An internet of things middleware for resource constrained mobile devices**. In *2014 47th Hawaii International Conference on System Sciences* (2014), IEEE, pp. 1053–1062.

[103] RAULT, T., BOUABDALLAH, A., AND CHALLAL, Y. **Energy efficiency in wireless sensor networks: A top-down survey**. *Computer Networks 67* (2014), 104 – 122.

[104] RAZZAQUE, M. A., AND DOBSON, S. **Energy-efficient sensing in wireless sensor networks using compressed sensing**. *Sensors 14*, 2 (2014), 2822–2859.

[105] ROLLASON, J. C., OUTTRIM, J. G., AND MATHUR, R. S. **A pilot study comparing the duofertility® monitor with ultrasound in infertile women**. *International journal of women's health 6* (2014), 657.

[106] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., AND SALAKHUTDINOV, R. **Dropout: A simple way to prevent neural networks from overfitting**. *Journal of Machine Learning Research 15* (2014), 1929–1958.

[107] YANG, L., LI, W., GE, Y., FU, X., GRAVINA, R., AND FORTINO, G. **People-Centric Service for mHealth of Wheelchair Users in Smart Cities**. Springer International Publishing, Cham, 2014, pp. 163–179.

[108] ABADI, M., AGARWAL, A., BARHAM, P., BREVDO, E., CHEN, Z., CITRO, C., CORRADO, G. S., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., GOODFELLOW, I., HARP, A., IRVING, G., ISARD, M., JIA, Y., JOZEFOWICZ, R., KAISER, L., KUDLUR, M., LEVENBERG, J., MANÉ, D., MONGA, R., MOORE, S., MURRAY, D., OLAH, C., SCHUSTER, M., SHLENS, J., STEINER, B., SUTSKEVER, I., TALWAR, K., TUCKER, P., VANHOUCKE, V., VASUDEVAN, V., VIÉGAS, F., VINYALS, O., WARDEN, P., WATTENBERG, M., WICKE, M., YU, Y., AND ZHENG, X. **TensorFlow: Large-scale machine learning on heterogeneous systems**, 2015. Software available from tensorflow.org.

[109] AL-FUQAHA, A., GUIZANI, M., MOHAMMADI, M., ALEDHARI, M., AND AYYASH, M. **Internet of things: A survey on enabling technologies, protocols, and applications**. *IEEE Communications Surveys Tutorials 17*, 4 (Fourthquarter 2015), 2347–2376.

[110] ANZANPOUR, A., RAHMANI, A.-M., LILJEBERG, P., AND TENHUNEN, H. **Internet of things enabled in-home health monitoring system using early warning score**. In *Proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare* (2015), pp. 174–177.

[111] BACÀ, A., BIAGETTI, G., CAMILLETTI, M., CRIPPA, P., FALASCHETTI, L., ORCIONI, S., ROSSINI, L., TONELLI, D., AND TURCHETTI, C. **Carma: A robust motion artifact reduction algorithm for heart rate monitoring from ppg signals**. In *2015 23rd European Signal Processing Conference (EUSIPCO)* (Aug 2015), pp. 2646–2650.

[112] BAGNALL, A., LINES, J., HILLS, J., AND BOSTROM, A. **Time-series classification with cote: The collective of transformation-based ensembles**. *IEEE Transactions on Knowledge and Data Engineering 27*, 9 (Sep. 2015), 2522–2535.

[113] CHEN, Y., KEOGH, E., HU, B., BEGUM, N., BAGNALL, A., MUEEN, A., AND BATISTA, G. **The ucr time series classification archive**, July 2015. www.cs.ucr.edu/~eamonn/time_series_data/.

[114] CHOLLET, F., AND OTHERS. **Keras**. https://keras.io, 2015.

[115] DUBEY, H., GOLDBERG, J. C., MANKODIYA, K., AND MAHLER, L. **A multi-smartwatch system for assessing speech characteristics of people with dysarthria in group settings**. *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)* (Oct 2015).

[116] FLORIDO, E., MARTÍNEZ-ÁLVAREZ, F., MORALES-ESTEBAN, A., REYES, J., AND AZNARTE-MELLADO, J. **Detecting precursory patterns to enhance earthquake prediction in chile**. *Computers & Geosciences 76* (2015), 112 – 120.

[117] GAETA, M., LOIA, V., AND TOMASIELLO, S. **Multisignal 1-d compression by f-transform for wireless sensor networks applications**. *Appl. Soft Comput. 30*, C (May 2015), 329–340.

[118] HE, K., ZHANG, X., REN, S., AND SUN, J. **Delving deep into rectifiers: Surpassing human-level performance on imagenet classification**. *CoRR abs/1502.01852* (2015).

[119] IOFFE, S., AND SZEGEDY, C. **Batch normalization: Accelerating deep network training by reducing internal covariate shift**. *CoRR abs/1502.03167* (2015).

[120] LAHAT, D., ADALI, T., AND JUTTEN, C. **Multimodal data fusion: an overview of methods, challenges, and prospects**. *Proceedings of the IEEE 103*, 9 (2015), 1449–1477.

[121] LEMIRE, D., AND BOYTSOV, L. **Decoding billions of integers per second through vectorization**. *Softw. Pract. Exper. 45*, 1 (Jan. 2015), 1–29.

[122] LIPTON, Z. C., KALE, D. C., ELKAN, C., AND WETZEL, R. **Learning to diagnose with lstm recurrent neural networks**. *arXiv preprint arXiv:1511.03677* (2015).

[123] MCCALL, W. V. **A rest-activity biomarker to predict response to ssris in major depressive disorder**. *Journal of psychiatric research 64* (2015), 19–22.

[124] OLLANDER, S. **Wearable sensor data fusion for human stress estimation**. Master's thesis, Technical University of Linkoping University, 2015.

[125] PHAN, N. H., DOU, D., PINIEWSKI, B., AND KIL, D. **Social restricted boltzmann machine: Human behavior prediction in health social networks**. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (08 2015).

[126] POLTAVSKI, D. V. **The use of single-electrode wireless eeg in biobehavioral investigations**. In *Mobile Health Technologies*. Springer, 2015, pp. 375–390.

[127] RAMSUNDAR, B., KEARNES, S., RILEY, P., WEBSTER, D., KONERDING, D., AND PANDE, V. **Massively multitask networks for drug discovery**. *arXiv preprint arXiv:1502.02072* (2015).

[128] RONNEBERGER, O., FISCHER, P., AND BROX, T. **U-net: Convolutional networks for biomedical image segmentation**. *CoRR abs/1505.04597* (2015).

[129] SERDAROGLU, K., USLU, G., AND BAYDERE, S. **Medication intake adherence with real time activity recognition on iot**. In *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (2015), pp. 230–237.

[130] SHARMA, R., AND PACHORI, R. B. **Classification of epileptic seizures in eeg signals based on phase space representation of intrinsic mode functions**. *Expert Systems with Applications 42*, 3 (2015), 1106–1117.

[131] SHIN, H.-C., LU, L., KIM, L., SEFF, A., YAO, J., AND SUMMERS, R. M. **Interleaved text/image deep mining on a very large-scale radiology database**. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1090–1099.

[132] SOHN, S. Y., BAE, M., LEE, D. R., AND KIM, H. **Alarm system for elder patients medication with iot-enabled pill bottle**. In *2015 International Conference on Information and Communication Technology Convergence (ICTC)* (2015), pp. 59–61.

[133] STOLLENGA, M. F., BYEON, W., LIWICKI, M., AND SCHMIDHUBER, J. **Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation**. In *Advances in neural information processing systems* (2015), pp. 2998–3006.

[134] TERVONEN, J., ISOHERRANEN, V., AND HEIKKILÄ, M. **A review of the cognitive capabilities and data analysis issues of the future industrial internet-of-things**. In *2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)* (Oct 2015), pp. 127–132.

[135] UKIL, A., BANDYOPADHYAY, S., AND PAL, A. **Iot data compression: Sensor-agnostic approach**. In *2015 Data Compression Conference* (April 2015), pp. 303–312.

[136] YI, C., WANG, L., AND LI, Y. **Energy efficient transmission approach for wban based on threshold distance**. *IEEE sensors journal 15*, 9 (2015), 5133–5141.

[137] ANGERMUELLER, C., LEE, H. J., REIK, W., AND STEGLE, O. **Accurate prediction of single-cell dna methylation states using deep learning**. *BioRxiv* (2016), 055715.

[138] AZARIADI, D., TSOUTSOURAS, V., XYDIS, S., AND SOUDRIS, D. **Ecg signal analysis and arrhythmia detection on iot wearable medical devices**. In *2016 5th International conference on modern circuits and systems technologies (MOCAST)* (2016), IEEE, pp. 1–4.

[139] AZIMI, I., RAHMANI, A. M., LILJEBERG, P., AND TENHUNEN, H. **Internet of things for remote elderly monitoring: a study from user-centered perspective**. *Journal of Ambient Intelligence and Humanized Computing 8* (06 2016).

[140] BAGNALL, A., BOSTROM, A., LARGE, J., AND LINES, J. **The great time series classification bake off: An experimental evaluation of recently proposed algorithms. extended version**, 2016.

[141] CHENG, L., GUO, S., WANG, Y., AND YANG, Y. **Lifting wavelet compression based data aggregation in big data wireless sensor networks**. In *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)* (Dec 2016), pp. 561–568.

[142] DI, S., AND CAPPELLO, F. **Fast error-bounded lossy hpc data compression with sz**. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (May 2016), vol. 00, pp. 730–739.

[143] DIMITROV, D. V. **Medical internet of things and big data in healthcare**. *Healthcare informatics research 22*, 3 (2016), 156–163.

[144] DUBEY, H., GOLDBERG, J. C., ABTAHI, M., MAHLER, L., AND MANKODIYA, K. **Echowear: Smartwatch technology for voice and speech treatments of patients with parkinson's disease**, 2016.

[145] FERATI, M., KURTI, A., VOGEL, B., AND RAUFI, B. **Augmenting requirements gathering for people with special needs using iot: A position paper**. In *2016 IEEE/ACM Cooperative and Human Aspects of Software Engineering (CHASE)* (2016), pp. 48–51.

[146] GONDARA, L. **Medical image denoising using convolutional denoising autoencoders**. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)* (2016), IEEE, pp. 241–246.

[147] GOODFELLOW, I., BENGIO, Y., AND COURVILLE, A. **Deep Learning**. The MIT Press, 2016.

[148] GRECO, A., VALENZA, G., LANATA, A., SCILINGO, E. P., AND CITI, L. **cvxeda: A convex optimization approach to electrodermal activity processing**. *IEEE Transactions on Biomedical Engineering 63*, 4 (April 2016), 797–804.

[149] HABIB, C., MAKHOUL, A., DARAZI, R., AND COUTURIER, R. **Multisensor data fusion for patient risk level determination and decision-support in wireless body sensor networks**. In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (2016), pp. 221–224.

[150] HUANG, G., LIU, Z., VAN DER MAATEN, L., AND WEINBERGER, K. Q. **Densely connected convolutional networks**, 2016.

[151] KOLDIJK, S., NEERINCX, M. A., AND KRAAIJ, W. **Detecting work stress in offices by combining unobtrusive sensors**. *IEEE Transactions on Affective Computing 9*, 2 (2016), 227–239.

[152] KOLDIJK, S. J. **Context-aware support for stress self-management: from theory to practice**. PhD thesis, [SI: sn], 2016.

[153] LANGARIZADEH, M., AND MOGHBELI, F. **Applying naive bayesian networks to disease prediction: a systematic review**. *Acta Informatica Medica 24*, 5 (2016), 364.

[154] METCALF, D., MILLIARD, S. T., GOMEZ, M., AND SCHWARTZ, M. **Wearables and the internet of things for health: Wearable, interconnected devices promise more efficient and comprehensive health care**. *IEEE pulse 7*, 5 (2016), 35–39.

[155] MOVASSAGHI, S., MAJIDI, A., JAMALIPOUR, A., SMITH, D., AND ABOLHASAN, M. **Enabling interference-aware and energy-efficient coexistence of multiple wireless body area networks with unknown dynamics**. *IEEE Access 4* (01 2016), 1–1.

[156] NEGRA, R., JEMILI, I., AND BELGHITH, A. **Wireless body area networks: Applications and technologies**. *Procedia Computer Science 83* (2016), 1274 – 1281. The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology (SEIT-2016) / Affiliated Workshops.

[157] ONG, B. T., SUGIURA, K., AND ZETTSU, K. **Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting pm 2.5**. *Neural Computing and Applications 27*, 6 (2016), 1553–1566.

[158] PIWEK, L., ELLIS, D. A., ANDREWS, S., AND JOINSON, A. **The rise of consumer health wearables: promises and barriers**. *PLoS medicine 13*, 2 (2016).

[159] RATHORE, M. M., AHMAD, A., PAUL, A., AND RHO, S. **Urban planning and building smart cities based on the internet of things using big data analytics**. *Comput. Netw. 101*, C (June 2016), 63–80.

[160] REAGEN, B., WHATMOUGH, P., ADOLF, R., RAMA, S., LEE, H., LEE, S. K., HERNÁNDEZ-LOBATO, J. M., WEI, G.-Y., AND BROOKS, D. **Minerva: Enabling low-power, highly-accurate deep neural network accelerators**. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)* (2016), IEEE, pp. 267–278.

[161] SHI, W., CAO, J., ZHANG, Q., LI, Y., AND XU, L. **Edge computing: Vision and challenges**. *IEEE Internet of Things Journal 3*, 5 (Oct 2016), 637–646.

[162] UKIL, A., BANDYOAPDHYAY, S., PURI, C., AND PAL, A. **Iot healthcare analytics: The importance of anomaly detection**. In *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)* (2016), pp. 994–997.

[163] VARGHESE, B., WANG, N., BARBHUIYA, S., KILPATRICK, P., AND NIKOLOPOULOS, D. S. **Challenges and opportunities in edge computing**. In *2016 IEEE International Conference on Smart Cloud (SmartCloud)* (Nov 2016), pp. 20–26.

[164] WANG, Z., YAN, W., AND OATES, T. **Time series classification from scratch with deep neural networks: A strong baseline**. *CoRR abs/1611.06455* (2016).

[165] XIONG, P., WANG, H., LIU, M., ZHOU, S., HOU, Z., AND LIU, X. **Ecg signal enhancement based on improved denoising auto-encoder**. *Engineering Applications of Artificial Intelligence 52* (2016), 194 – 202.

[166] **National early warning score (news), royal college of physicians, london, u.k., may 2015.** https://www.rcplondon.ac.uk/projects/outputs/national-early-warning-score-news, 2017. Accessed: 2020-04-27.

[167] AMOR, L. B., LAHYANI, I., AND JMAIEL, M. **Pca-based multivariate anomaly detection in mobile healthcare applications**. In *2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real Time Applications (DS-RT)* (2017), IEEE, pp. 1–8.

[168] BOUDARGHAM, N., ABDO, J. B., DEMERJIAN, J., AND GUYEUX, C. **Exhaustive study on medical sensors**. In *SENSORCOMM 2017, The Eleventh International Conference on Sensor Technologies and Applications* (2017), IARIA.

[169] CHEN, M., HAO, Y., HWANG, K., WANG, L., AND WANG, L. **Disease prediction by machine learning over big data from healthcare communities**. *Ieee Access 5* (2017), 8869–8879.

[170] COLLET, Y. **Zstandard-fast real-time compression algorithm**. https://facebook.github.io/zstd/, 2017.

[171] CRESWELL, A., ARULKUMARAN, K., AND BHARATH, A. A. **On denoising autoencoders trained to minimise binary cross-entropy**. *CoRR abs/1708.08487* (2017).

[172] DUA, D., AND GRAFF, C. **UCI machine learning repository**, 2017.

[173] EGEA, S., MAÑEZ, A. R., CARRO, B., SÁNCHEZ-ESGUEVILLAS, A., AND LLORET, J. **Intelligent iot traffic classification using novel search strategy for fast-based-correlation feature selection in industrial environments**. *IEEE Internet of Things Journal 5*, 3 (2017), 1616–1624.

[174] HABIB, C., MAKHOUL, A., DARAZI, R., AND COUTURIER, R. **Real-time sampling rate adaptation based on continuous risk level evaluation in wireless body sensor networks**. In *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (2017), IEEE, pp. 1–8.

[175] HU, J., SHEN, L., ALBANIE, S., SUN, G., AND WU, E. **Squeeze-and-excitation networks**, 2017.

[176] KWAN, J. C., AND FAPOJUWO, A. O. **Radio frequency energy harvesting and data rate optimization in wireless information and power transfer sensor networks**. *IEEE Sensors Journal 17*, 15 (2017), 4862–4874.

[177] LEA, C., FLYNN, M. D., VIDAL, R., REITER, A., AND HAGER, G. D. **Temporal convolutional networks for action segmentation and detection**. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017).

[178] MAINETTI, L., MIGHALI, V., PATRONO, L., RAMETTA, P., AND STEFANIZZI, M. L. **An iot-aware system for elderly monitoring**. In *2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)* (2017), pp. 1–5.

[179] MALASINGHE, L., RAMZAN, N., AND DAHAL, K. **Remote patient monitoring: a comprehensive study**. *Journal of Ambient Intelligence and Humanized Computing* (10 2017).

[180] MIGHALI, V., PATRONO, L., STEFANIZZI, M. L., RODRIGUES, J. J., AND SOLIC, P. **A smart remote elderly monitoring system based on iot technologies**. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)* (2017), IEEE, pp. 43–48.

[181] PARK, S. J., SUBRAMANIYAM, M., KIM, S. E., HONG, S., LEE, J. H., JO, C. M., AND SEO, Y. **Development of the elderly healthcare monitoring system with iot**. In *Advances in Human Factors and Ergonomics in Healthcare*. Springer, 2017, pp. 309–315.

[182] ROY, V., SHUKLA, S., SHUKLA, P. K., AND RAWAT, P. **Gaussian elimination-based novel canonical correlation analysis method for eeg motion artifact removal**. *Journal of healthcare engineering 2017* (2017).

[183] SAEED, A., AND TRAJANOVSKI, S. **Personalized driver stress detection with multi-task neural networks using physiological signals**. *arXiv preprint arXiv:1711.06116* (2017).

[184] SAEED, A., TRAJANOVSKI, S., VAN KEULEN, M., AND VAN ERP, J. **Deep physiological arousal detection in a driving simulator using wearable sensors**. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)* (2017), IEEE, pp. 486–493.

[185] SATYANARAYANAN, M. **The emergence of edge computing**. *Computer 50*, 1 (Jan 2017), 30–39.

[186] SHEN, H., GEORGE, D., HUERTA, E. A., AND ZHAO, Z. **Denoising gravitational waves using deep learning with recurrent denoising autoencoders**, 2017.

[187] STOJKOSKA, B. R., AND NIKOLOVSKI, Z. **Data compression for energy efficient iot solutions**. In *2017 25th Telecommunication Forum (TELFOR)* (Nov 2017), pp. 1–4.

[188] UM, T. T., PFISTER, F. M. J., PICHLER, D., ENDO, S., LANG, M., HIRCHE, S., FIETZEK, U., AND KULIC, D. **Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks**. *CoRR abs/1706.00527* (2017).

[189] ZUHRA, F. T., BAKAR, K. A., AHMED, A., AND TUNIO, M. A. **Routing protocols in wireless body sensor networks: A comprehensive survey**. *Journal of Network and Computer Applications 99* (2017), 73–97.

[190] ALIEKSIEIEV, V. **One approach of approximation for incoming data stream in iot based monitoring system**. In *2018 IEEE Second International Conference on Data Stream Mining Processing (DSMP)* (Aug 2018), pp. 94–97.

[191] ANTCZAK, K. **Deep recurrent neural networks for ecg signal denoising**. *ArXiv abs/1807.11551* (2018).

[192] AZAR, J., DARAZI, R., HABIB, C., MAKHOUL, A., AND DEMERJIAN, J. **Using dwt lifting scheme for lossless data compression in wireless body sensor networks**. In *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)* (2018), IEEE, pp. 1465–1470.

[193] AZAR, J., HABIB, C., DARAZI, R., MAKHOUL, A., AND DEMERJIAN, J. **Using adaptive sampling and dwt lifting scheme for efficient data reduction in wireless body sensor networks**. In *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (2018), IEEE, pp. 1–8.

[194] AZAR, J., MAKHOUL, A., DARAZI, R., DEMERJIAN, J., AND COUTURIER, R. **On the performance of resource-aware compression techniques for vital signs data in wireless body sensor networks**. In *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)* (2018), IEEE, pp. 1–6.

[195] BAGNALL, A. J., DAU, H. A., LINES, J., FLYNN, M., LARGE, J., BOSTROM, A., SOUTHAM, P., AND KEOGH, E. J. **The UEA multivariate time series classification archive, 2018**. *CoRR abs/1811.00075* (2018).

[196] BLALOCK, D., MADDEN, S., AND GUTTAG, J. **Sprintz: Time series compression for the internet of things**. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 2*, 3 (Sept. 2018), 93:1–93:23.

[197] CASTANEDA, D., ESPARZA, A., GHAMARI, M., SOLTANPUR, C., AND NAZERAN, H. **A review on wearable photoplethysmography sensors and their potential future applications in health care**. *International Journal of Biosensors and Bioelectronics 4*, 4 (2018), 195–202.

[198] CHOI, H.-S., KIM, S., OH, J. E., YOON, J. E., PARK, J. A., YUN, C.-H., AND YOON, S. **Xgboost-based instantaneous drowsiness detection framework using multitaper spectral information of electroencephalography**. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics* (2018), pp. 111–121.

[199] COUTURIER, R., PERROT, G., AND SALOMON, M. **Image denoising using a deep encoder-decoder network with skip connections**. In *Neural Information Processing* (Cham, 2018), L. Cheng, A. C. S. Leung, and S. Ozawa, Eds., Springer International Publishing, pp. 554–565.

[200] ELSAYED, N., MAIDA, A. S., AND BAYOUMI, M. **Deep gated recurrent and convolutional network hybrid model for univariate time series classification**. *CoRR abs/1812.07683* (2018).

[201] FAWAZ, H. I., FORESTIER, G., WEBER, J., IDOUMGHAR, L., AND MULLER, P. **Deep learning for time series classification: a review**. *CoRR abs/1809.04356* (2018).

[202] GANDHI, S., OATES, T., MOHSENIN, T., AND HAIRSTON, D. **Denoising time series data using asymmetric generative adversarial networks**. In *Advances in Knowledge Discovery and Data Mining* (Cham, 2018), D. Phung, V. S. Tseng, G. I. Webb, B. Ho, M. Ganji, and L. Rashidi, Eds., Springer International Publishing, pp. 285–296.

[203] HARB, H., MAKHOUL, A., AND JAOUDE, C. A. **En-route data filtering technique for maximizing wireless sensor network lifetime**. In *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)* (June 2018), pp. 298–303.

[204] HARB, H., MAKHOUL, A., AND JAOUDE, C. A. **A real-time massive data processing technique for densely distributed sensor networks**. *IEEE Access 6* (2018), 56551–56561.

[205] HARTH, N., ANAGNOSTOPOULOS, C., AND PEZAROS, D. **Predictive intelligence to the edge: impact on edge analytics**. *Evolving Systems 9*, 2 (Jun 2018), 95–118.

[206] KARIM, F., MAJUMDAR, S., DARABI, H., AND CHEN, S. **Lstm fully convolutional networks for time series classification**. *IEEE Access 6* (2018), 1662–1669.

[207] KARIM, F., MAJUMDAR, S., DARABI, H., AND HARFORD, S. **Multivariate lstm-fcns for time series classification**. *CoRR abs/1801.04503* (2018).

[208] LE, T. L., AND VO, M. **Lossless data compression algorithm to save energy in wireless sensor network**. In *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)* (Nov 2018), pp. 597–600.

[209] LI, H., OTA, K., AND DONG, M. **Learning iot in edge: Deep learning for the internet of things with edge computing**. *IEEE Network 32*, 1 (Jan 2018), 96–101.

[210] ORUN, A., ELIZONDO, D., GOODYER, E., AND PALUSZCZYSZYN, D. **Use of bayesian inference method to model vehicular air pollution in local urban areas**. *Transportation Research Part D: Transport and Environment 63* (2018), 236–243.

[211] PROCESSING, A. D. **The workforce view in europe 2018**. https://www.leslivresblancs.fr/livre/entreprise/ressources-humaines/workforce-view-europe-2018, 2018. Accessed: 2020-04-06.

[212] RAGUSA, E., GIANOGLIO, C., GASTALDO, P., AND ZUNINO, R. **A digital implementation of extreme learning machines for resource-constrained devices**. *IEEE Transactions on Circuits and Systems II: Express Briefs 65*, 8 (2018), 1104–1108.

[213] SEHATBAKHSH, N., ALAM, M., NAZARI, A., ZAJIC, A., AND PRVULOVIC, M. **Syndrome: Spectral analysis for anomaly detection on medical iot and embedded devices**. In *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* (2018), pp. 1–8.

[214] SINGHA ROY, M., GUPTA, R., CHANDRA, J. K., DAS SHARMA, K., AND TALUKDAR, A. **Improving photoplethysmographic measurements under motion artifacts using artificial neural network for personal healthcare**. *IEEE Transactions on Instrumentation and Measurement 67*, 12 (Dec 2018), 2820–2829.

[215] TAYEH, G. B., MAKHOUL, A., DEMERJIAN, J., AND LAIYMANI, D. **A new autonomous data transmission reduction method for wireless sensors networks**. In *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)* (2018), IEEE, pp. 1–6.

[216] TAYEH, G. B., MAKHOUL, A., LAIYMANI, D., AND DEMERJIAN, J. **A distributed real-time data prediction and adaptive sensing approach for wireless sensor networks**. *Pervasive and Mobile Computing 49* (2018), 62–75.

[217] VAN GENT, P., FARAH, H., VAN NES, N., AND VAN AREM, B. **Analysing noisy driver physiology real-time using off-the-shelf sensors: Heart rate analysis software from the taking the fast lane project**. *Journal of Open Research Software 7*, 1 (2018), 32.

[218] ZAREEI, S., AND DENG, J. D. **Impact of compression ratio and reconstruction methods on ecg classification for e-health gadgets: A preliminary study**. In *AI 2018: Advances in Artificial Intelligence* (Cham, 2018), T. Mitrovic, B. Xue, and X. Li, Eds., Springer International Publishing, pp. 85–97.

[219] ALANSARI, Z., ANUAR, N. B., KAMSIN, A., SOOMRO, S., AND BELGAUM, M. R. **Evaluation of iot-based computational intelligence tools for dna sequence analysis in bioinformatics**. In *Progress in Advanced Computing and Intelligent Engineering*. Springer, 2019, pp. 339–350.

[220] ASKARI, M. R., RASHID, M., SEVIL, M., HAJIZADEH, I., BRANDT, R., SAMADI, S., AND CINAR, A. **Artifact removal from data generated by nonlinear systems: Heart rate estimation from blood volume pulse signal**. *Industrial & Engineering Chemistry Research* (2019).

[221] EKIENABOR, E. **Impact of job stress on employees' productivity and commitment**. *International Journal for Research in Business, Management and Accounting 2* (07 2019), 124–133.

[222] FAWAZ, H. I., FORESTIER, G., WEBER, J., IDOUMGHAR, L., AND MULLER, P.-A. **Adversarial attacks on deep neural networks for time series classification**. In *2019 International Joint Conference on Neural Networks (IJCNN)* (2019), IEEE, pp. 1–8.

[223] FAWAZ, H. I., FORESTIER, G., WEBER, J., IDOUMGHAR, L., AND MULLER, P.-A. **Deep neural network ensembles for time series classification**. In *2019 International Joint Conference on Neural Networks (IJCNN)* (2019), IEEE, pp. 1–6.

[224] FAWAZ, H. I., LUCAS, B., FORESTIER, G., PELLETIER, C., SCHMIDT, D. F., WEBER, J., WEBB, G. I., IDOUMGHAR, L., MULLER, P.-A., AND PETITJEAN, F. **Inceptiontime: Finding alexnet for time series classification**. *arXiv preprint arXiv:1909.04939* (2019).

[225] FERRY, K. **Workplace stress continues to mount**. https://www.kornferry.com/insights/articles/workplace-stress-motivation, 2019. Accessed: 2020-04-06.

[226] KWON, D., KIM, H., KIM, J., SUH, S. C., KIM, I., AND KIM, K. J. **A survey of deep learning-based network anomaly detection**. *Cluster Computing 22*, 1 (Jan 2019), 949–961.

[227] LI, S., AND ZHANG, X. **Research on orthopedic auxiliary classification and prediction model based on xgboost algorithm**. *Neural Computing and Applications* (2019), 1–9.

[228] LIN, J.-W., LIAO, S.-W., AND LEU, F.-Y. **Sensor data compression using bounded error piecewise linear approximation with resolution reduction**. *Energies 12*, 13 (2019), 2523.

[229] MELKI, R., NOURA, H. N., AND CHEHAB, A. **Lightweight and secure d2d authentication & key management based on pls**. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)* (2019), IEEE, pp. 1–7.

[230] NOURA, H., COUTURIER, R., PHAM, C., AND CHEHAB, A. **Lightweight stream cipher scheme for resource-constrained iot devices**. In *2019 International*

Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) (2019), IEEE, pp. 1–8.

[231] RICHARD, G., HÉBRAIL, G., MOUGEOT, M., AND VAYATIS, N. **Densenets for time series classification: towards automation of time series pre-processing with cnns**. In *DenseNets for Time Series Classification: towards automation of time series pre-processing with CNNs* (2019).

[232] SARBISHEI, O. **Refined lightweight temporal compression for energy-efficient sensor data streaming**. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* (2019), IEEE, pp. 550–553.

[233] SHARMA, R. R., KUMAR, A., PACHORI, R. B., AND ACHARYA, U. R. **Accurate automated detection of congestive heart failure using eigenvalue decomposition based features extracted from hrv signals**. *Biocybernetics and Biomedical Engineering 39*, 2 (2019), 312–327.

[234] VAN GENT, P., FARAH, H., VAN NES, N., AND VAN AREM, B. **Heartpy: A novel heart rate algorithm for the analysis of noisy signals**. *Transportation Research Part F: Traffic Psychology and Behaviour 66* (2019), 368 – 378.

[235] WLODARCZAK, P. **Springer: Deep learning in ehealth**. In *Handbook of Deep Learning Applications*. Springer, 2019, pp. 319–331.

[236] **Biopac systems france**. https://www.biopac.com/product-category/research/fnir-optical-brain-imaging/sensor-options/, 2020. Accessed: 2020-04-27.

[237] **Shimmer sensing**. http://www.shimmersensing.com/, 2020. Accessed: 2020-04-27.

[238] AWAD, N., COUCHOT, J.-F., AL BOUNA, B., AND PHILIPPE, L. **Publishing anonymized set-valued data via disassociation towards analysis**. *Future Internet 12*, 4 (2020), 71.

[239] YAACOUB, J.-P. A., NOURA, M., NOURA, H. N., SALMAN, O., YAACOUB, E., COUTURIER, R., AND CHEHAB, A. **Securing internet of medical things systems: Limitations, issues and recommendations**. *Future Generation Computer Systems 105* (2020), 581–606.

[240] BAYDOGAN, M. **Mustafa baydogan - multivariate time series classification data sets**. http://www.mustafabaydogan.com/files/viewcategory/20-data-sets.html. Accessed: 2020-02-04.

[241] T.S, L. **Biometric human identification based on electrocardiogram**. Master's thesis, Faculty of Computing Technologies and Informatics, Electrotechnical University "LETI", Saint-Petersburg, Russian Federation, June 2005.

List of Figures

List of Tables

Document generated with LaTeX and:
the LaTeX style for PhD Thesis created by S. Galland — http://www.multiagent.fr/ThesisStyle
the tex-upmethodology package suite — http://www.arakhne.org/tex-upmethodology/

école doctorale **sciences pour l'ingénieur et microtechniques**

Title: Data Compression and Deep Learning for IoT Healthcare Applications based on Physiological Signals

Keywords: Energy Consumption, Data compression, Digital signal processing, Internet of Things, Stress detection, Physiological signals, Deep learning

Abstract:

In recent years, Internet of Things (IoT) technology has gained tremendous attention for its ability to relieve the burden on healthcare caused by an aging population and the increase in chronic disease. IoT technology facilitates the tracking of patients with different conditions and the processing of vast volumes of data, of which a substantial part of this data are physiological signals. Physiological signals are an invaluable source of data which helps to diagnose, rehabilitate, and treat diseases. The signals come from a Wireless Body Sensor Network or wearable devices placed on a patient's body. There are many difficulties in IoT-healthcare systems, such as data collection and processing.

This dissertation mainly focuses on the data collection as well as data processing and analysis approaches for wireless body sensor networks and IoT-healthcare. To do so, two main axes have been considered, namely, the data collection process of an IoT-healthcare application and edge-level data analysis and processing. The first axis tackles the energy-efficiency and the problem of finding a trade-off between compression ratio and classification performance. The second axis studies the use of deep learning techniques to solve numerous challenges, such as physiological time series reconstruction, enhancement, and classification.

Titre : Data Compression and Deep Learning for IoT Healthcare Applications based on Physiological Signals

Mots-clés : Consommation d'énergie, Compression de données, Traitement numérique du signal, Internet des objets, Signaux physiologiques, Deep learning

Résumé :

Ces dernières années, la technologie de l'Internet des objets (IoT) a suscité un vif intérêt en raison de sa capacité à alléger certaines tâches de soins de santé causé par le vieillissement de la population et l'augmentation des maladies chroniques. La technologie IoT facilite le suivi des patients souffrant de différentes pathologies et le traitement de vastes volumes de données, dont une partie substantielle de ces données sont des signaux physiologiques. Les signaux physiologiques sont une source inestimable de données qui aident à diagnostiquer, réhabiliter et traiter les maladies. Les signaux proviennent d'un réseau de capteurs corporels sans fil ou d'appareils portables placés sur le corps d'un patient. Il existe de nombreuses difficultés dans les systèmes de soins de santé IoT, tels que la collecte et le traitement des données.

Cette thèse se focalise principalement sur la collecte de données ainsi que sur les approches de traitement et d'analyse des données pour les réseaux de capteurs corporels sans fil et les systèmes de soins de santé IoT. Dans cet ordre d'idées, deux axes principaux ont été considérés, à savoir le processus de collecte de données d'une application IoT-santé et l'analyse et le traitement des données au niveau de la périphérie (edge). Le premier axe aborde l'efficacité énergétique et le problème de trouver un compromis entre le taux de compression et les performances de classification. Le deuxième axe étudie l'utilisation des techniques d'apprentissage en profondeur pour résoudre de nombreux défis, tels que la reconstruction, l'amélioration et la classification de séries temporelles physiologiques.

UBFC UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

Université Bourgogne Franche-Comté
32, avenue de l'Observatoire
25000 Besançon, France