Diplomarbeit

XAI for human-centred production: a practical application by the example of HAR

ausgeführt um Zwecke der Erlangung des akademischen Grades

Diplom-Ingenieur

eingereicht an der

Technische Universität Wien
Fakultät für Informatik / Maschinenwesen und Betriebswissenschaften

unter der Leitung von:
Univ.-Prof. Dr.-Ing Sebastian Schlund
(Institut für Managementwissenschaftern, Forschungsbereich: Mensch-Maschine-Interaktion)

und
Dipl.-Ing. David Kostolani
(Institut für Managementwissenschaftern, Forschungsbereich: Mensch-Maschine-Interaktion)

von
Alejandro Manchado Rubio, BSc.

Matrikelnummer: 12019243

[redacted]

[redacted]

Wien, 28.02.2022

Alejandro Manchado Rubio

TECHNISCHE UNIVERSITÄT WIEN

Ich habe zur Kenntnis genommen, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

XAI for human-centred production: a practical application by the example of HAR

nur mit Bewilligung der Prüfungskommission berechtigt bin.

Ich erkläre weiters Eides statt, dass ich meine Diplomarbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen selbstständig ausgeführt habe und alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, genannt habe. Weiters erkläre ich, dass ich dieses Diplomarbeitsthema bisher weder im In- noch Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe und dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Wien, 28.02.2022

Alejandro Manchado Rubio

Abstract

Data processing technologies such as machine learning have emerged and become increasingly complicated as the complexity of modern products and manufacturing processes has increased. Model complexity precludes finding possible problems since they are often treated as black boxes. This was also coupled with the recent growth in concern about privacy and cybersecurity, as well as the influence of legislation emphasising the necessity of data protection, such as the European GDPR. In this context, the demand for transparency and interpretability for machine learning algorithms has risen.

This thesis argues for the importance of interpretability in machine learning models, especially in a human-centred production environment. Furthermore, methodologies currently in use for human activity recognition (HAR) were analysed in the literature. Different approaches for improving the explainability and interpretability of commonly used models are also highlighted. Finally, a repeatable methodology is proposed in this research to enhance the recognition of human activities.

The proposed technique is divided into two parts: the first proposal involves data preparation and the use of the LIME and Submodular-Pick LIME explication algorithms in order to increase model interpretability. The interpretable results of the first proposal are used in the second suggested practise to reduce the amount of information introduced into the model. The results show that there is a trade-off between model's accuracy in recognising human activities and the privacy of user data. However, applying the proper techniques the detection accuracy remains high, even though 55% of the data is removed. This enhances user privacy and leads to the use of less invasive models for the worker.

Key words: Human Activity Recognition; Explainable Artificial Intelligence; XAI; Interpretable Machine Learning; LIME;

Kurzfassung

Aufgrund der steigenden Komplexität moderner Produkte und Fertigungsprozesse kommen immer kompliziertere Technologien, wie maschinelles Lernen, zur Verarbeitung der Daten zum Einsatz. Die Komplexität der verwendeten Modelle erschwert allerdings das Entdecken möglicher Probleme bei der Verarbeitung, da diese Modelle oft als eine "Black Box" behandelt werden. Hinzu kommt, dass die Bedeutung der Themen wie der Schutz der Privatsphäre und die Cybersicherheit in den letzten Jahren zugenommen hat. Die Notwendigkeit des Datenschutzes wird auch von Rechtsvorschriften, wie beispielsweise die Europäische Datenschutz-Grundverordnung, betont. In diesem Zusammenhang ist die Nachfrage nach Transparenz und Interpretierbarkeit für Algorithmen des maschinellen Lernens gestiegen.

Diese Arbeit beschäftigt sich mit der Interpretierbarkeit von Modellen des maschinellen Lernens und hebt die Bedeutung der Interpretierbarkeit in einer menschenzentrierten Produktionsumgebung hervor. Hierzu wurde die Literatur über die derzeit verwendeten Methoden für HAR (Human Activity Recognition - Erkennung menschlicher Aktivitäten) analysiert. Weiters wurden verschiedene Ansätze zur Verbesserung der Erklärbarkeit und der Interpretierbarkeit von häufig verwendeten Modellen aufgezeigt. Schließlich wird in dieser Arbeit eine reproduzierbare Methodik vorgeschlagen, um die Erkennung menschlicher Aktivitäten zu verbessern.

Die vorgeschlagene Methodik gliedert sich in zwei Teile: Der erste Schritt umfasst die Datenvorbereitung und die Verwendung der Explikationsalgorithmen LIME und Submodular-Pick-LIME, um die Interpretierbarkeit der Modelle zu erhöhen. Anhand der Interpretation der Modelle wird im nächsten Schritt die Anzahl der dem Modell verabreichten Daten reduziert. Die Ergebnisse zeigen, dass es einen Kompromiss zwischen der Genauigkeit der Erkennung menschlicher Aktivitäten und der Privatsphäre der Benutzerdaten gibt. Bei der Anwendung der richtigen Techniken bleibt die Erkennungsgenauigkeit jedoch hoch, selbst wenn 55% der Daten entfernt werden. Durch diese Methodik werden Modelle weniger invasiv, was sich in einer Stärkung der Privatsphäre der NutzerInnen widerspiegelt.

Key words: Erkennung menschlicher Aktivitäten; Erklärbare künstliche Intelligenz; erklärbare KI; interpretierbares maschinelles Lernen; LIME;

Acknowledgements

My sincere gratitude to my supervisor, David Kostolani, for helping me to complete this thesis. From the beginning, he has been a huge supporter of my work. This work would not have been possible without his regular feedback as well as proactive and additional support in the implementation of the thesis.

I am also thankful for the opportunities to learn new things that this research has provided me. In addition, I could greatly enhance my understanding of machine learning, particularly explainable machine learning.

At this point, I would like to express gratitude to my lifelong friends who have always supported me. More specifically, I would like to thank all the friends I have made during this master's degree at the Technical University of Vienna, J. Sebastian, Sergi, Sofia, Pablo, Marina, Maria C., Maria F., Ivan, Isabel, Guillem, Dana, Carol, Belen, Anna, Ana, and Ainara who made this stay an unforgettable experience.

I would also like to thank my girlfriend Sara, for all her love and support.

Finally, I would want to express my gratitude to my family, particularly my parents. My Master's would not have been possible without their unwavering support.

Contents

1. Introduction

This proposed study seeks to advance knowledge on the need for Explainable Artificial Intelligence in Assistance Systems and the benefits it would bring to this field. To justify the need, the reason and present difficulties that motivate such implementation will first be examined, and the intended results of this research will be accompanied by the techniques required to obtain them.

1.1. Motivation and Problem Statement

Due to the issue of increased product and process complexity, manufacturing organizations with a strong focus on assembly processes are experiencing a need for transformation. Short development periods, individualization of client needs, flexibility, decentralization, and acceleration of fulfilment operations are all examples of how process complexity is increasing [1]. Additionally, demographic change adds to the difficulty of process management concerns.

However, in industrial contexts, new innovative technologies are influencing the new generation of assembly systems based on the notion of cyber-physical systems (CPS). CPSs are systems of collaborative computational entities that are in close proximity to the physical world and its ongoing processes, providing and utilizing data-processing and data-accessing services at the same time [2].

A new type of manufacturing will be achievable by tightly integrating CPS into a production system. These previously mentioned challenges in the industry are expected to be met by the so-called Cyber-Physical Production Systems. The effectiveness of CPPS is dependent in part on the accuracy with which technologies can recognize context in order to provide adaptive feedback on a specific sequence of activities, ergonomics, or the proper use of tools. Employee assistance systems should be human-centred technical platforms that provide them with real-time feedback [3].

Implicit Interaction and Human Activity Recognition (HAR) will be key methods for incorporating context-awareness into Human-Centred CPPS and assistance systems. The goal of HAR is to automatically classify activities based on data collected from several sensing modalities. This process is usually performed using machine learning algorithms [4]. However, often the problem with the machine learning algorithms used in these systems is that they are essentially “black boxes”, since it is hard to understand how the data are processed. However, in many situations, we cannot afford to sacrifice interpretability. Interpretability has no consensus around its definition due to its subjective nature; however, it can be associated with understandability, accuracy, and transparency [5].

The need for transparency and interpretability in machine learning models or explainable artificial intelligence (XAI) is widely recognized in many studies in the literature for a variety of applications such as medical diagnoses [6] or recommender systems [7], [8], because it provides multiple benefits, for example, extracting interpretable patterns from trained models; identifying reasons for poor decisions; increasing confidence in model decisions; aiding in the detection of bias in machine learning models; adding a safety net to protect against overfitted models [9]. However, the use of XAI for assistance systems in the production sector is currently not explored sufficiently.

The use of XAI in assistance systems is critical to safety and security. The increasing complexity and connectivity of Cyber-Physical Systems, as well as the tight coupling between their cyber and physical components and the inevitability of human operators being involved in their supervision control, have posed significant challenges in ensuring system reliability and safety while maintaining expected performance. In all engineering disciplines, the phrase "safety" refers to the absence of faults or situations that make a system unsafe. In machine learning, we define safety as the reduction of both risk and epistemic uncertainty associated with undesirable outcomes severe enough to be considered harmful [10]. Epistemic uncertainty arises from a lack of knowledge that could theoretically be obtained but is difficult to obtain in practice.

CPS constantly interact with the physical world and human operators in real-time. As a result, they must consider not only the present application but also the manufacturer's preferences, purpose, and previous behaviour in order to adapt to the constantly changing and uncertain environment. The danger arises when we make or use decisions that are not justifiable or legitimate, particularly in applications where experts require more information from the model than simple binary predictions, such as heavy machinery handling, autonomous vehicle transportation, security, finance, etc.

In the context of machine learning, robustness is desirable against the uncertainty of the training set not being sampled from the test distribution. The training set may contain biases that the user is unaware of, or patterns that could lead to negative effects. And because machine learning models are complicated, it is impossible to predict how they will react to changes in the data domain. One of the solutions defined in [10] to improve robustness and achieve safety in Cyber-Physical Systems, as well as other common present applications, is to build an inherently safe model. That is, by using models that can be interpreted and omitting features that are not causally related to the outcome. In this way, quirks in the data can be identified and excluded, preventing the harm that comes with them.

Furthermore, in recent years, there has been a growing public awareness of privacy and cybersecurity. On the one hand, this is due to numerous data leaks. Over the past few years, the amount of data breaches and information released has increased, some of them being as significant as Facebook data exposures [11]. On the other hand, legislation has an impact on the importance of data protection. In 2018, the European Union passed the General Data Protection Regulation (GDPR), which establishes strict guidelines

for the use and storage of personal data. At the same time, this European regulation expands automated decision-making rights to include a legally contested version of a "right to an explanation." The right to an explanation is a mathematical regulation that refers to an individual's right to be informed about actions that have a major impact on them, especially legally or financially. It is stated in this way in Recital 71 of the GDPR: "[the data subject should have] the right ... to obtain an explanation of the decision reached" [12]. As a result, interpretable models are required to comprehend and safely handle industrial data in order to select just the data that is strictly required for the given function, as otherwise private information about the worker, task, products, and firm could be leaked.

In addition to the legal considerations that have already been discussed, ethical concerns must also be taken into account. The algorithm should include respect for basic human rights, such as dignity, equality, and nondiscrimination, to address all ethical concerns [13]. For example, the most significant ethical values described in the Beijing Principles are the following: In the part dedicated to Research and Development "Do Good", "Be Responsible", "Open and Share", "Be Diverse and Inclusive" or "Be Ethical". Principles of a technical or operational nature such as "Optimizing Employment", "Adaptation and Moderation", "Subdivision and Implementation" or "Long-term Planning" are covered in the section devoted to Governance [14].

In this scenario, the interpretability of the model helps us to confirm that the system meets all ethical standards during auditing. Meaning that our support system should operate for everyone, regardless of their traits (sex, skin colour, etc.). Anything that could physically or emotionally hurt people should be avoided or minimized. It should also make sure that participants are aware of any potential risks before participating while making an effort to remain objective and neutral. Allowing your personal biases or ideas to influence the data collection process is not a good idea.

This master's thesis will further analyse the use of XAI using the example of Human Activity Recognition. For this specific use case, certain work task can possess a high level of similarities, which might hinder correct recognition. If recognition does not work for these tasks, the only way to improve the AI system is to understand how decisions are made and which features contribute the most to the recognition of the work task. In other words, the interpretability of the model also allows for further analysis of where the system is failing or to justify why one decision was made over another.

1.2. Expected Outcome

The main result that is expected to be achieved from the theoretical part of this research is the expansion of the current state of knowledge on the topics of assistance systems, machine learning, and AI explainability. In order to answer the first research question: **What is interpretability in machine learning and why is it important for human-centred production?** Evidence in favour of the usage of interpretable machine

learning algorithms in human activity recognition will be provided. For this purpose, among other outcomes, this study also aims to obtain findings and recommendations on the strategies to achieve interpretability in machine learning models.

Furthermore, the review of the state of the art literature will serve as a baseline for the practical implementation, providing detailed review on the methods currently in use. This review will also provide an analysis of the explainability and interpretability of the models in order to obtain an answer for the second research question. **How can the explainability and interpretability of models typically used for HAR be achieved?**

Finally, with the expected results from the practical part, the last research question will be answered: **How can XAI help to improve the design of HAR models for use in work assistance systems?** For this, an example of improvement of these models will be carried out in which essentially accuracy measures in HAR by various ML algorithms are expected to be obtained, as well as an assessment of which algorithm performs better, and which inputs contribute the most and least to recognition. The purpose of this is to evaluate whether the algorithm can work similarly with a comparable accuracy if the least important inputs are omitted. As a consequence of all of this, this research intends to obtain a realistic and repeatable methodology to improve Human Activity Recognition tasks, in which similar IMUs are used. And its implementations would be encouraged in order to achieve the best performance while preserving the privacy of worker data to the greatest extent possible.

1.3. Methodology

This research aims to support and justify the use of explainable AI in HAR algorithms for the industry so that worker data can be used more efficiently. Several procedures will be followed in order to answer the research questions previously mentioned in chapter 2. For example, a review of the literature will be carried out to answer the theoretical part of this study. Numerous sources from the literature will be studied to be able to describe classic machine learning algorithms and neural networks in detail. After that, a comparison with regard to interpretability and explainability between the various models will be made. Allowing us to answer the research question of which models are the most relevant for our particular application. This information will prove useful later when deciding which algorithms to use for the practical aspect of the investigation.

Furthermore, in this study, a comprehensive analysis of the current state of the art will be conducted, which will focus on two aspects. On the one hand, because IMU data will be employed later in the practical phase of the research, a state of the art review on inertial sensors for HAR, particularly those used for support systems or manufacturing applications, will be conducted. And we will also examine the advances made in the state of the art in this field in terms of the privacy of data collected by workers' sensors.

On the other hand, due to the necessity to use interpretable models in this research, a second part of a review of the state of the art will be provided in this research. The analysis should go through the explainability and interpretability in further detail. We will concentrate on obtaining a thorough understanding of how interpretability and explainability can be achieved in a model. And what are the most suitable metrics to evaluate the quality of an explanation. Finally, the last part of the research will be more practical and will consist of the implementation of several classification models, among them classic machine leaning models, such as search trees. The performance of these models against neural networks, specifically Recurrent Neural Networks (RNN) and Feedforward neural networks (FNN), will be compared, measuring the correlation between the input features and the output variable, as well as the contribution of every input to the recognition. This will be done by measuring the precision of the model when inferring data with a larger or smaller number of input variables. In addition, if it is concluded that there are any inputs that are less important for the recognition, as a result, they might be left out in order to utilize fewer data to do the same task while maintaining the worker's privacy as much as possible.

The Python programming language will be used to implement these methods, as it provides compact and legible code in the programming of machine learning models. In addition, the machine learning library "scikit-learn" will be used, which includes various classification, regression, and clustering algorithms, in collaboration with the library "LIME", used to explain any machine learning black box classi-fier with two or more classes. Also, Keras, a high-level neural network API will be used, as it also is user-friendly for deep learning.

Another important material for carrying out this research is the UCI HAR dataset on "Human Activity Recognition Using Smartphones." This contains data on human activities (walking, walking down-stairs and upstairs, standing, sitting, and laying) gathered utilizing embedded IMU sensors, i.e., the accelerometer and the gyroscope, of a smartphone on the waist of 30 volunteers to obtain triaxial data of linear acceleration and angular velocity.

2. Theoretical Foundations

2.1. Cyber Physical Production Systems

Within the past few decades, the role of computers in the workplace has evolved considerably. While a corporation might have had a single mainframe computer years ago, which cost several millions of dollars and required an entire computing centre to run, computers have become common office equipment over time. Each user now has a number of additional microprocessors installed in daily objects in addition to their own computer [15].

Manufacturing is one of the most significant applications of automated systems. Nowadays, the ultimate goal of every manufacturing company is to continually produce quality products. However, the achievement of this objective is affected by the recent appearance of some challenges, which were already mentioned in the introduction section 1.1 Motivation and Problem Statement, regarding to market, technological, organizational and human-resource requirements. In order to avoid defects during an increasing number of variants in manual production processes, to meet the competition from low-cost companies, and to secure globally consistent quality, current production processes are defined by a very high degree of automation.

However, with shorter product life cycles, an increasing need for product variations, and complex production strategies used by many companies, full automation of all production processes, leaving out human operations, seems to be not feasible [16]. Excessive automation might lead to poor system performance [17]. Additionally, advanced production processes are prone to disruptions generally. As a result, human workers will continue to be a valuable source of production. Despite the high level of industrialisation in the industry, the human operator is frequently a component of the production system, and its participation in technical developments is required for flexible and efficient manufacturing [18].

In summary, in order to efficiently manage production facilities, design processes and equipment, plan and control production orders, and meet product quality standards, more interactive electronic support systems are becoming necessary for production [19]. Manufacturing assistance systems are the collection of procedures of a company's to manage production and resolve technical and logistical issues in assembly and disassembly, information management, training, and inspection [20]. Although most of these support systems do not interact directly with the product, they plan and manage its journey through the plant. Production processes become noticeably more trustworthy and efficient as a result of this, and the requirement to provide information regarding product outcomes is ensured.

Furthermore, in today's production and production management, so-called cyber-physical systems (CPS) are becoming more common [21]. CPSs are defined by the fact that the physical and software components are closely linked. In these systems, embedded sensors in manufacturing facilities and goods gather contextual information in real time and provide personalized process support in intelligent and multi-modal assistance systems.

Sensorics, among other benefits, provides a higher degree of automated collection and processing. It also broadens management clarity across the supply chain. Clarity helps companies enhance operational efficiency, since it helps reduce idle time, optimizing tasks, etc. Consequently, embedded sensors have helped optimize the performance of manufacturing equipment, leading to greater efficiency and productivity gains.

2.1.1. Motivation for Activity Recognition in Production

One of the tasks that has been affected by the emergence of this type of sensors is Human Activity Recognition (HAR). In the past, collecting sensor data for activity detection was difficult and expensive, requiring custom hardware. Smartphones and other personal tracking devices for health and fitness monitoring, which use this type of sensor, are now widely available and inexpensive. As a result, sensor data from these devices are less expensive to obtain and more frequent, making them a better investigated variant of the overall activity identification problem.

The objective of Human Activity Recognition (HAR) should be to build a model that predicts the current task or behaviour. The existence of technology capable of accurately categorizing a user's physical activity is very appealing for a wide range of applications [22]. Because the Industry 4.0 movement is not migrating towards workerless manufacturing facilities, HAR is a highly common task in Cyber Physical Systems in order to incorporate humans into the CPS [23]. By human activities, we refer to those such as "sitting", "walking", and "running," which arise very naturally in daily life and are relatively easy to recognize. On the other hand, more complex activities are more difficult to identify. However, complex activities may be decomposed into other simpler activities, which are generally easier to recognize [24].

In a human-centred intelligent manufacturing system, sensing and understanding of the worker's activity are the primary tasks. This information is a valuable resource, since it may be employed to quantify and evaluate anonymized performance indicators such as the mean time of each work task, etc. Also, recognition of worker activity is crucial for human-robot interaction and collaboration [25]. Therefore, it is essential to develop intelligent human-centred manufacturing systems [26].

The monitoring of human activities may be done in two ways: sensor-based and vision-based. Despite the fact that both of these approaches are non-intrusive and will have no effect on human activities, the sensor-based way of gathering data from individuals has

been proven to be better than the other. Because this non-intrusive technique interferes less with people's privacy. Moreover, sensor-based monitoring is immune to extraneous disturbances that might confuse and distort the data obtained. Therefore, it has been shown to be more suitable [27].

Furthermore, as previously stated, there are many low-cost wearable gadgets on the market, such as smart armbands and smartphones, that are frequently utilized in activity identification activities [26]. Human body's movement can be instantly detected by wearable technology, such as a wristband with an Inertial Measurement Unit (IMU). This technology may even offer health information. IMUs sensors are commonly used in Human Activity Recognition (HAR) tasks [3]. Occlusion is not a problem with wearable gadgets because they are directly linked to the human body. However, because a wearable device can only detect the movement of parts of the human body, it is difficult to properly distinguish an action involving several regions at the same time. Therefore, multiple sensors are often necessary to detect global activity [26].

Wearable sensors have advanced to the point that they can now measure parameters in a continuous, real-time, and non-intrusive manner. In particular, IMUs are inertial sensors that combine gyroscopes, accelerometers, and, in many cases, magnetometers to measure velocity, orientation, and gravitational forces. These sensors are also commonly found in inertial navigation systems found in airplanes, spacecrafts, and other vehicles, but they are of special interest when tracking the positions or postures of people who wear them while performing tasks [28].

2.2. Fundamentals of Machine Learning

Humans are not the only beings capable of learning. Learning behaviour may be demonstrated by other species and even artificial systems. Since the beginning of Artificial Intelligence (AI) research, envisioning thinking and learning robots has piqued curiosity (1950). Interest in artificial intelligence has been reignited as a result of recent advances in machine learning algorithms in a variety of fields [29].

Artificial intelligence (AI) has grown in popularity both inside and outside the scientific community over the past decade. There are numerous papers in both technical and non-technical journals covering the themes of machine learning (ML), deep learning (DL), and AI [30]. The widespread gathering of data using electronic methods, as a result of increased internet use or the availability of low-cost sensors, has resulted in an exponentially expanding volume of "big data." As a result, the amount of digital data available has become too large to handle [31]. Machine learning and, more recently deep learning, are two major approaches that have proven the capacity to turn huge datasets into usable information. However, there is still a lot of misunderstanding about AI, machine learning, and deep learning. Although the terms are closely related, they are not interchangeable.

Machine learning, as described by Arthur Samuel in 1959, is the "area of research that enables computers to learn without being explicitly programmed." Later, the Well-posed Learning Problem was established by Tom Mitchell (1998): "A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance in T, as measured by P, improves with experience E."

"Every component of learning or any other attribute of intelligence [might], in principle, be so accurately defined that a machine [could] be created to imitate it," a group of computer scientists claimed in 1956. This principle was dubbed "artificial intelligence" by them. Simply defined, AI is a discipline dedicated to automating intellectual work that would otherwise be handled by people, and ML and DL are two ways to accomplish this aim. That is, AI incorporates them (Fig. 2.1) [30][32][33]. AI, on the other hand, also encompasses other methods that do not rely on "learning." However, more difficult tasks are where ML and DL techniques stand out.

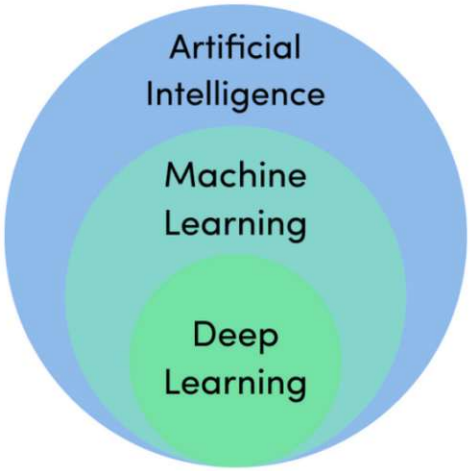


Figure 2.1.: Machine learning (ML) is included within the concept of Artificial Intelligence (AI). Although ML contains many models and methods, including deep learning (DL).
Source: [30]

ML is a subset of artificial intelligence that focuses on the learning element of technology by creating algorithms that best represent a collection of data. ML trains using huge data sets to build an algorithm that may use unique or different combinations of weights and features than can be determined from first principles, as opposed to classical programming, where the software routines were hand-coded with a specific set of instructions [30][32][33][34]. In machine learning, there are three major learning methods: supervised, unsupervised, and reinforcement learning, each of which is effective in tackling distinct tasks.

The machine learns on a labelled dataset in a supervised learning model, which provides an answer key that the program may use to evaluate its training accuracy.

On the other hand, an unsupervised model gives unlabelled data that the system tries to gain understanding of for itself by identifying patterns. And reinforcement learning uses a reward mechanism to train an algorithm, giving feedback when the artificial intelligence algorithm takes the best action in a given circumstance [34].

Supervised Learning

Predicting property prices is an example of supervised learning issues in practice. Using specific features of the building, such as number of rooms, square meters, and other features, such as if there is a garden on the property, etc. The values of these houses, i.e. targets, are then required for the supervised machine in order to train to estimate the price of a new property based on the instances observed by the model. Another popular problem is image classification. To predict, for example, if the animal in the image is a dog or a cat, we first would have needed to feed classified training samples to the computer.

As it was mentioned in the previous example, the value to be predicted is the target. Having a complete set of labelled data while training an algorithm is required in supervised learning. It would work as if someone were there when you are learning a task under supervision, evaluating if you are obtaining the correct response [34]. Training, validation, and testing datasets are the most common ways to split the datasets. The training set is used to improve model parameters, while the cross-validation set is used to choose the optimal model among those with different hyperparameters. Eventually, on the test dataset, which contains data that the model has not seen during previous processes, the algorithm' s performance may be computed [35].

The basic steps of supervised machine learning are to: (1) obtain a dataset and divide it into training, validation, and test; (2) use the training and validation datasets to compute the parameters of an algorithm using the relationship between features and target; and (3) evaluate the model using the test dataset to see how well it predicts the target for unknown instances. The performance of the algorithm on the training data is compared with the performance in the validation dataset in each iteration. The validation set is used to tweak the algorithm in this way [30][32][33].

Regression and classification are two of the most frequent supervised learning problems. As in the preceding example of house price prediction, regression entails predicting continuous data. On the other hand, classification requires estimating a discrete value, recognising the input data as belonging to a specific class or group, corresponding to the categorization of animals described in the preceding example. The system is then assessed on the basis of how well it can categorize new data. As a result, supervised learning is particularly well suited to issues with a collection of existing reference points or ground truth for the algorithm to learn from [30][32][33][34].

Unsupervised Learning

Datasets that are clean and correctly labelled are hard to come by. On occasion, the

algorithm aims to solve questions for which the researchers have no answers. Unsupervised learning comes into play in this situation, and, in contrast to supervised learning, the model tries to find patterns in a dataset with no clear instructions and categorize individual occurrences into some classes. These algorithms are unsupervised, since the algorithm is left to discover the patterns that may or may not exist in a dataset with no clear desired outcome or right response [30][32][33][34].

Clustering, anomaly detection, and association are some of the most frequent unsupervised learning tasks.

- In **Clustering**, which is the most common application for unsupervised learning, the model divides samples from a dataset into distinct groups depending on similar combinations of their characteristics.

- With **Anomaly detection**, unsupervised learning may also be used to identify odd trends or outliers in a dataset.

- Finally, **Association** is based on certain characteristics of a data sample that are linked to others. This allows an unsupervised learning model to predict the other qualities with which a data point is often linked by looking at a few key features of the data point.

It is difficult to assess the quality of an algorithm learned with unsupervised learning, since the data lack a "ground truth" attribute. However, labelled data is difficult to come by in many study areas, or it is too expensive. In some instances, allowing the deep learning model to discover patterns on its own can yield excellent results[34].

Reinforced Learning

Finally, reinforcement learning is a strategy for teaching a machine learning algorithm to achieve a given objective or enhance performance in a specific job in which not just one response is correct, but a desirable overall result is sought. A reward is given to the model as it takes steps toward the objective. Because it learns through trial and error rather than just data, it is perhaps the closest mimic to the human learning process [30][32][33][34].

The agent bases its decisions on prior feedback as well as the search for different techniques that may offer a higher return. However, this necessitates a long-term strategy; just as the greatest immediate move in a chess game may not help you win in the long run, the model seeks to maximize the total pay-off. It is an iterative process: the more feedback rounds the algorithm receives, the better its approach is [34].

This method is particularly beneficial for training robots that must make a succession of judgements in activities such as autonomous car driving. Also, it is used to train an algorithm to play a video game like Mario Bros. In this last example, there is no

good or bad input sequence. In reinforcement learning, an algorithm would be allowed to "play" on its own. It would experiment with various controller inputs until it successfully carried Mario forward (without harming him), at which time the algorithm would be "rewarded." [30].

Performance Evaluation in Supervised Learning

After the training phase has been completed, in supervised learning, the trained model's performance may be measured in a variety of ways, but the most frequent are, for regression algorithms, error and residuals, and prediction accuracy for classification algorithms. The aim of an ML model is to learn how to achieve that output for new data instead of remembering the data that were shown throughout the training. Therefore, the test dataset, as stated above, is used to check to see if it performs well in unknown cases that have not been used for training [30].

It is not common that the function completely matches the dataset. Therefore, the residues, which are the vertical distances between actual y values and the predicted ones, $\hat{y}$, are used to calculate the error associated with a regression algorithm. The cost function is a calculus-derived concept, using the residuals values, that is used to evaluate the performance of the model [30][32][33].

Minimizing the cost function frequently returns parameter values that optimally fit a dataset. The objective of all ML algorithms is to reduce the cost function in order to discover the most accurate model. This minimization of the cost function is generally carried out using gradient descent, which is an iterative optimization technique [30][32][33].

For a classification model, the accuracies on the training and validation datasets are frequently used to assess performance of a model. And, as long as the model's performance in the training and validation set rises and converges after each iteration of the algorithm, it is considered to continue learning. Most of the time, the cause of poor performance in machine learning on a dataset is either a high variance or high bias problem. For example, if a model's prediction is accurate on the training dataset but inaccurate on the test dataset, the model is overfitted to the training dataset [30] [36].

When a model fails to function well even on the training samples and also fails to extrapolate to new ones, it is underfitting the data. A machine learning model that is underfit is unsuitable, as evidenced by its poor performance. It is unable to identify the correlation between the input and output examples. This phenomenon is also referred to as high bias, but is rarely mentioned since, it means that the model is too basic to accurately represent the output. Therefore, experimenting with different ML models with increased model flexibility can enhance performance [36] [37].

However, if a model works excessively accurately on the training data, but not on the test data, it is referred to as overfitting or high variance. It occurs when a model detects and learns the complexity and noise or random oscillations in the training data to the

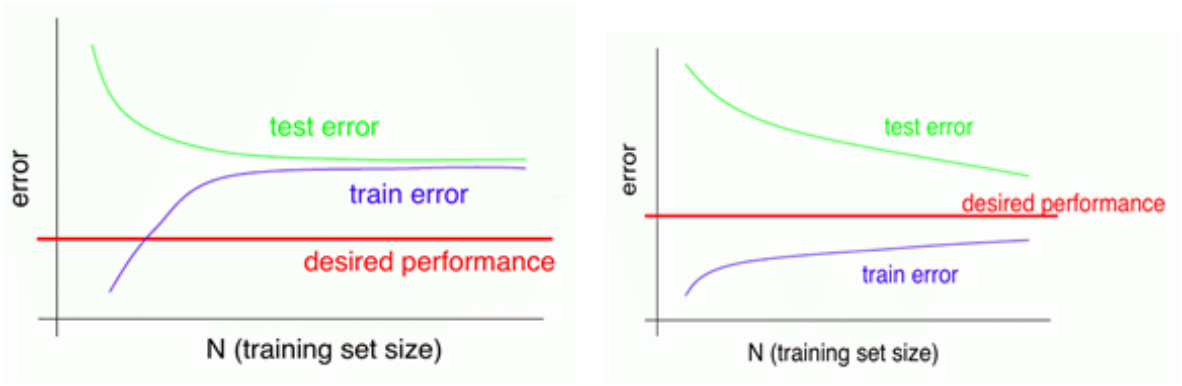
point where it worsens the model's performance in new data. This makes the model unable to extrapolate for instances that it has not previously observed, and therefore lowering the model's generalization potential. In contrast to the preceding case, in order to solve an overfitting scenario, it seems reasonable to take steps that decrease model flexibility. As a result, many non-parametric machine learning methods, which are more flexible, incorporate parameters or strategies to restrict the level of precision learned by the model [36] [37].

Also, accuracy on training and test data might be low because the learning algorithm does not have enough data to learn from. In that case, the solution could be increasing the number of instances of training data. Increasing the number of passes on the existing training data could also help improve the performance [37].

Understanding the model fit is crucial to figuring out what is causing substandard prediction performance. By comparing the estimation error on the testing and training data, we can tell whether a predictive model is underfitting or overfitting the training data. The plot of learning curves when utilizing machine learning might diagnose if the ML model has bias or variance by looking at those curves (see Fig. 2.2) [37].

In an underfitting situation, as N increases, both errors converge to the same asymptote as we can see in Fig. 2.2 (a), leaving no gap between the two curves. However, the error rate is much higher than the intended level. When the model is in a high bias situation expanding the training set will not help. Because of the model's biased assumptions, in this case obtaining more data would not solve the problem but adding more parameters could [35].

On the other hand, for the second picture, because the polynomial has a high degree in cases with overfitting, the training error will rise slowly but remain well within the required performance. However, it results in significant cross-validation errors (see Fig. 2.2 (b)). In this case, more data could be beneficial because the quantity of training data grows, therefore, the model is forced to learn more examples that an overfit curve cannot compensate for. As seen in Figure 2.2 (b), as the amount of training data grows, the gap between training and cross-validation error shrinks [35].



(a) (b)

Figure 2.2.: (a) High Bias and (b) High Variance Learning Curves
Source: [38]

Ideally, a model that works accurately on both datasets should be aimed for, which would be at the optimum between underfitting and overfitting (see Fig. 2.3) [36].

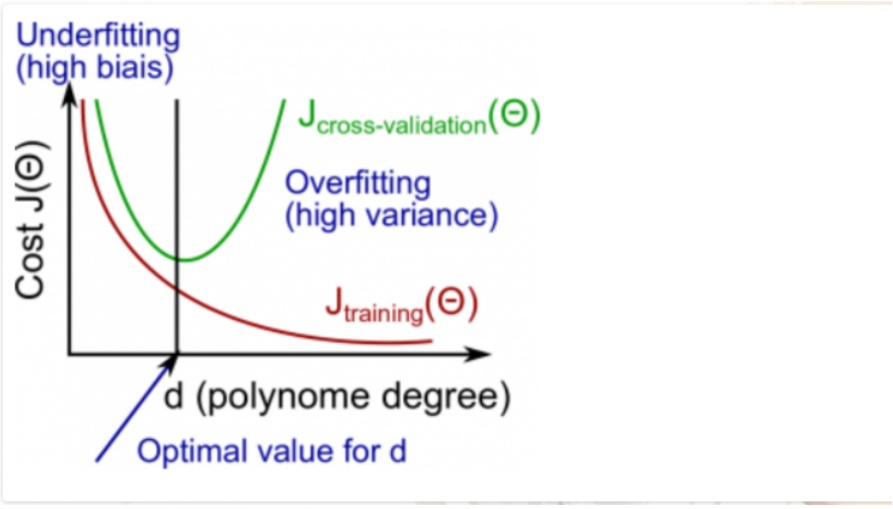


Figure 2.3.: Estimation of the optimal value for the degree of the polynomial (lambda) using the cost function J.
Source: [38]

2.2.1. Classical Supervised Algorithms in Machine Learning

Despite the availability of more complex algorithms, due to the parsimony principle, which states that the easiest solution that can explain the data should be selected, classic ML algorithms will continue to have a strong presence. Linear and Logistic Regression, Decision Trees, Support Vector Machines, and K-nearest Neighbours are just a few of the Classical ML Algorithms discussed in this area.

Linear Regression

The simplest machine learning technique is linear regression. The basic goal is to define a connection between some variables, but in this case the parameters used to describe the dataset are the ones from the equation of a straight line:

$$\hat{y} = ax + b \tag{2.1}$$

In equation (2.1), a is the slope of the straight line i.e., increment on the y-axis for each one in the x-axis. In this case, a is a weight that represents the slope, or how much a line grows on the y-axis for each increase in x. The place where the line intersects the y-axis is designated by letter b. In the case of multivariate linear regression, the technique equation includes several weights, each of which describes the degree to which each factor impacts the goal [30][32][33].

Logistic Regression

Logistic regression is a categorization method whose objective is to discover a link between features and the likelihood of a specific result. Logistic regression estimates class probability using a sigmoidal curve, that transforms discrete or continuous numeric characteristics (x) into a single numerical value (y) between 0 and 1, rather than the straight line previously generated by linear regression. The main benefit of this technique, in contrast to the previous algorithm (see Fig. 2.4) is that probabilities are limited to a range of 0 to 1. Logistic regression is frequently used as a preliminary step for binary classification applications due to its simplicity. The model can be either binomial or multinomial, meaning that it can classify into two or more possible classes [30][39].

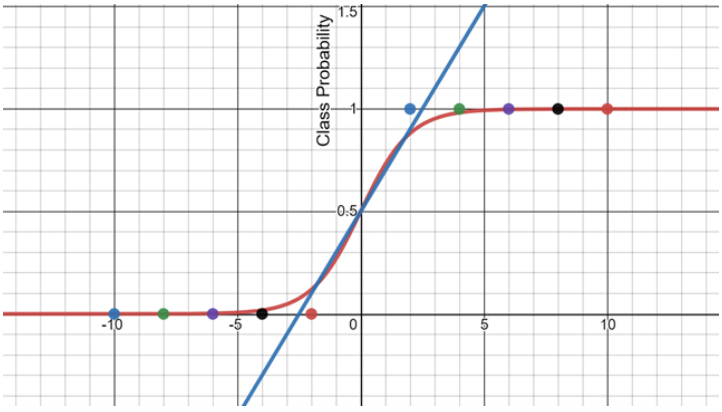


Figure 2.4.: Linear (blue) and logistic (red) regression representations for predicting the class probability. While, in linear regression the probability estimation does not have neither upper nor lower bound, in logistic regression, thanks to the sigmoid function, the values are limited between 0 and 1.

Decision Trees

A decision tree is a non-parametric supervised learning approach that may be used either for regression or classification tasks. The algorithm starts with a root node, which is the starting point of choice for the recursive partitioning that binarily divides the dataset into classes (Fig. 2.5). A single condition best splits the data which can link to either a terminal node that forecasts the class or a new decision node to further divide the data into more groups. The weights and the number of branches are established during the training phase [30][39].

A random forest, also known as an ensemble method, is an evolution of this technique that creates several decision trees. Also, instead of utilizing every feature to build every decision tree, a subset of features is utilized [30].

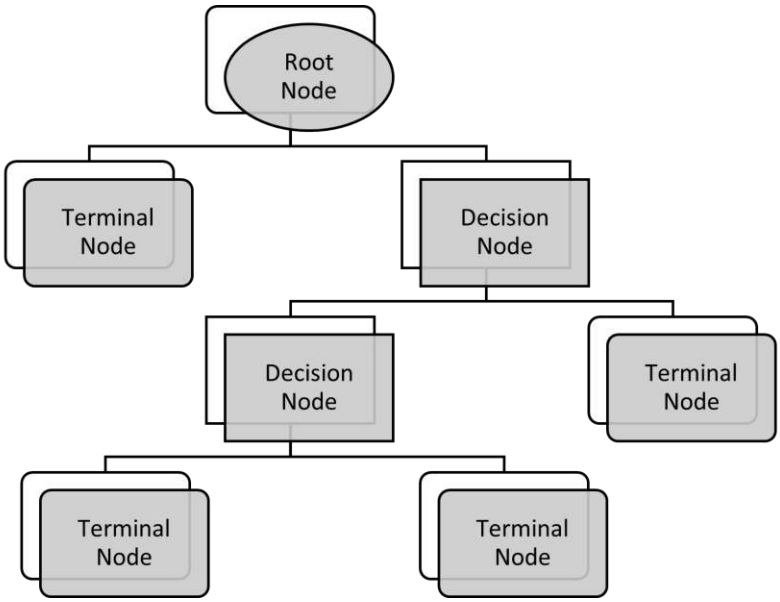


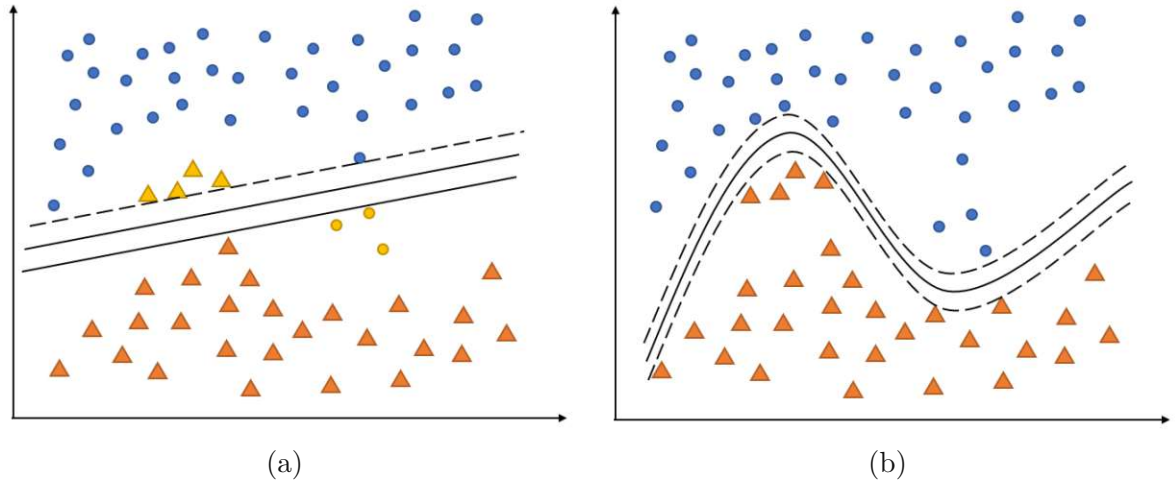
Figure 2.5.: The figure shows a decision tree's structure. The root node is where the dataset is split. Each split can link to another decision node, which splits the data even more, or to a terminal node already, which forecasts the data's category.

Support vector machine

Support vector machine (SVM) is also a supervised learning model used either for classification and regression. As many other algorithms, the SVM explicitly takes into account the fact that only the points nearest to the border are of importance for categorization. The ones that are really far away are easier to categorise and therefore have fewer impact. The task of this algorithms is to locate the data points that are closest to the boundary, referred to as "support vectors." The support vectors will be responsible for defining the optimal hyperplane, which is the one that divides both classes with the

greatest distance between them. After the accurate border is found, new inputs can be placed into the divided space. Each of the zones corresponds to a forecast category [39][40][41][42].

Only in situations with linear separation the method is able to locate this hyperplane; in most real situations, the technique optimizes the soft margin by tolerating a limited number of anomalies. The conventional SVM algorithm is designed to solve dichotomous classification problems, but multiclass problems can also be solved if they are simplified into several binary problems [40]. An example of dataset split by a Support Vector Machine hyperplane is shown in Fig. 2.6.



(a) (b)

Figure 2.6.: (a) linear and (b) non-linear Support Vector Machine visualization example for two-dimensional dataset

K-Nearest Neighbour (KNN)

The K-Nearest Neighbour (KNN) algorithm is another nonparametric technique for assigning a class label to the input pattern. It categorizes elements assuming that close objects are similar, and the item is attributed to the class k with the K-nearest neighbour. This is a commonly used algorithm due to the computational simplicity of the model and the positive results when using it with small sample sizes [39][43]. An example of this type of algorithm can be seen in Fig. XX.

However, one of the issues with utilizing the KNN algorithm is that every sample is generally given the same weight when assigning the class label. Therefore, abnormal vectors are given the same weight as those that are true cluster representatives, and there could be problems in areas where the sample sets intersect [43].

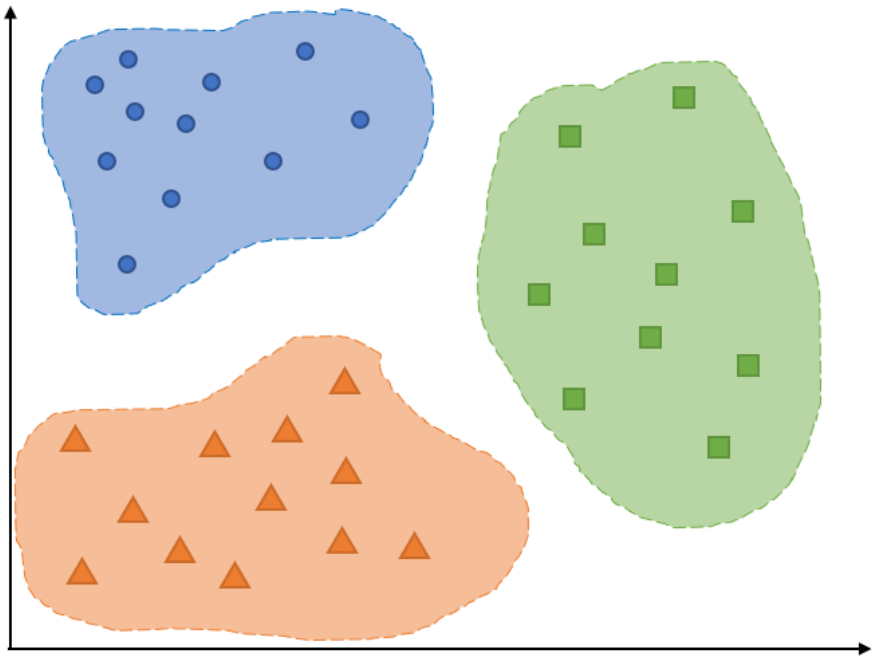


Figure 2.7.: The figure shows a K-Nearest Neighbour two-dimensional example. The algorithm has divided the dataset into two clusters represented in different colours.

2.2.2. Neural Networks

Machine learning, as explained previously, is the ability of computers to execute tasks that they have not been explicitly programmed to do. However, the capacity of classical algorithm to do some complicated tasks, such as image or video recognition, is still considerably inferior to that of humans.

Deep learning models, on the other hand, bring an exceptionally efficient strategy to machine learning and are suitable to solve these difficulties. In these algorithms data is transferred between nodes in highly linked pathways using complex, multi-layered "deep neural networks." As a result, the data undergoes several non-linear transformations.

Artificial neural networks, or just Neural Networks, are a type of deep learning model that consists of a group of components called artificial neurons that are connected to send signals. The input data is sent through the neural network, where it is exposed to different processes and output values are generated [44].

Each neuron is linked to the others by connections, in which the preceding neuron's output value is multiplied by a weight value, which enhances or inhibits the activity status of the neighbouring neurons. Similarly, there may be a limiting or threshold function, also known as the activation function, at the neuron's output that affects the outcome value [44].

Rather than being explicitly coded, these systems are self-trained, and are especially applied in areas where solution detection is difficult to be expressed through conventional programming. The ML model usually aims to reduce the loss function that assesses the network. Therefore, during training, the connection weights of the neurons are adjusted in an attempt to minimize the loss function's value during learning in a process called backpropagation. The gradient of the cost function associated with a given state with respect to the weights is computed throughout backpropagation. After that, using stochastic gradient descent, the weights may be adjusted [45].

As it was already mentioned, Neural networks have been used to tackle a number of problems that are difficult to handle with traditional rule-based programming, such as object recognition and language processing. Historically, weak NN-like architectures with few phases have been in use for a long time. Architectures with multiple nonlinear layers of neurons date back to approximately the 1960s and 1970s [46].

In this chapter, among the numerous types of neural networks, the feedforward neural network (FNN), the convolutional neural network (CNN), which are widely used in image and video recognition, and the recurrent neural network (RNN), which generally includes the long short-termed memory (LSTM), used in robotics and machine translation, will be covered.

Feedforward Neural Network

A feed-forward neural network (FNN) was the first and most basic type of ANN [47] in which the interconnections between the neurons do not create a loop or a cycle, but the layers are densely connected. In this model the data can only travel forward, from the input nodes to the output ones, passing through the hidden layer(s), if any (see Fig. 2.7).

Input data, the model architecture, a feedback mechanism so that the model learns, and a model training technique are all required to construct a feedforward NN. Regarding the model architecture, it consists of layers and nodes, and activation functions. Layers and nodes determine how complicated will be the Neural Network. All the neurons from each layer are completely connected with every node from the previous and next layer. Therefore, the more layers and nodes, the higher the capacity of the model [45].

Hidden layers, as we can see in Figure 5 are the ones in between the input and the output, they will determine the computational cost of the algorithm. The goal of the algorithm architecture design is to obtain the simplest model that can solve the problem with the best performance [45].

On the end or right-side of the figure are the output layers, which are determined by the use we want for the model. For example, the output layer will only be consisting of one node for regression problems, which will give a continuous number as output, while

in multiclass classification tasks, there will be the same number of output neurons as classes [45].

Activation or transfer functions are also a crucial feature of the FNN's architecture. Each neuron's connection is assigned a weight, which is multiplied by the input of that connection. Then all the values are summed and inserted into the activation function, which evaluates if the node has sufficient informative input to trigger a signal to the next layer [45].

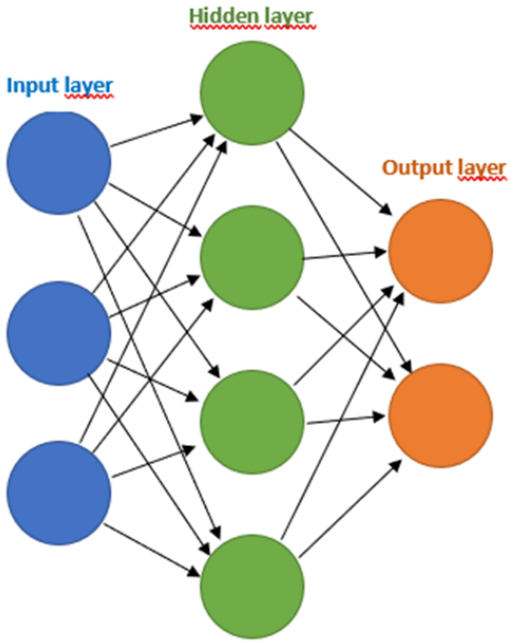


Figure 2.8.: Schematic structure of a Feed-forward Neural Network is shown, which presents only one hidden layer. In can be seen in this figure that the network of this model type is densely connected i.e., all the neurons are connected with every neuron in the consecutive layer.

It is also needed to mention the feedback mechanism used, the backpropagation. This means that, to carry out the learning mechanism, the FNN will assign randomly weights to all neuron connection and predict the outcome. Then it will evaluate the prediction and adjust the weights to attempt to improve it. To do this objective, a loss function is required. For classification problems the loss function commonly is categorical cross-entropy, and for regression problems, the mean squared error (MSE) [48][45].

Once the performance of the forward pass is evaluated using the chosen objective function, the FNN will compute the gradient in relation to the network weights to adjust them in a direction contrary to the gradient, until the loss function is minimized [45].

Because of FNNs' ability to approximate complex connections effectively from input samples and to give proper models for a large class of complex data, otherwise difficult to handle using traditional techniques, FNNs have been widely used in many fields [49].

Traditionally, all parameters of the feedforward network (weights and biases) had to be tuned for the application. Gradient descent-based approaches have mostly been utilized in various feedforward neural network learning algorithms for years. However, because this process is typically slow and requires many iterative cycles, other methods can be used instead [49].

Convolutional Neural Network

A convolutional neural network (CNN) is a subtype of artificial neural network. It is a regularized variant of a multilayer perceptron, which is a form of FNN. Multilayer perceptrons are densely interconnected structures, which means that each neuron in one layer of the network is linked to all neurons in the next layer. However, this "density" makes them vulnerable to overfitting.

However, because deep learning models are often fed with two-dimensional matrices, such as images, unlike a conventional neural network, CNN's layers have neurons organised in three dimensions: width, height, and depth. This architecture is highly successful for artificial vision tasks such as image classification and segmentation, among other uses. For time series, one-dimensional Convolutions are frequently employed, since in such cases input data is also one dimensional.

A three-dimensional depiction of a convolutional layer is shown in Fig. 2.9. The three-dimensional weight matrix in the centre of the figure is the kernel. To extract useful features, the kernel is multiplied by the input. In two-dimensional convolutions, the kernel is a two-dimensional matrix, whereas in one-dimensional CNNs, the kernel is one-dimensional. A filter, on the other hand, is a concatenation of numerous kernels, each of which is allocated to a certain input channel [50].

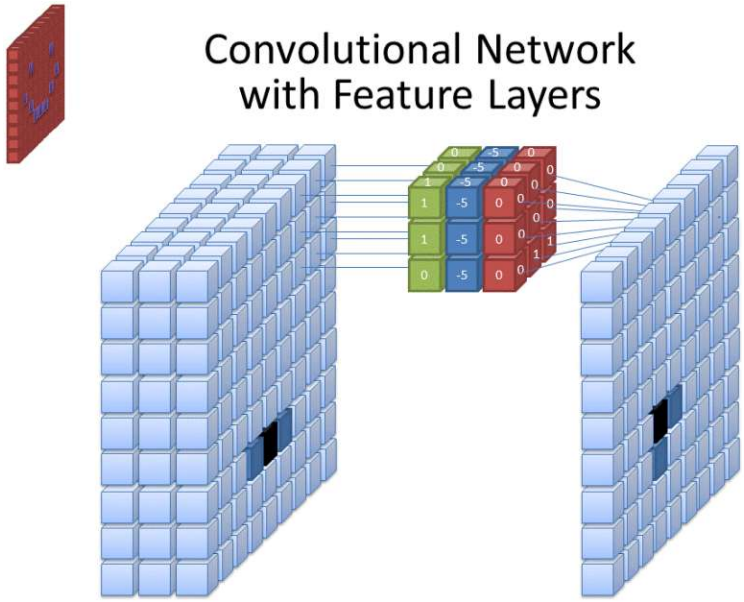


Figure 2.9.: Convolutional layer. Three feature maps are on the layer on the left, and the layer on the right is formed by applying the filter bank to the preceding one. The weights are shared among all the neurons in the same map, and the filter bank is the cube in the middle (on the right). The weights will be applied to each feature map from the layer on the left from each layer in the filter bank that has a distinct colour. Every neuron in the succeeding layer processes a patch of previous units using the filter map we weights, and the filter map advances across the unit of the previous layer.
Source: [51]

To prevent overfitting, dense neural network present typical ways of regularization, such as, penalizing parameters during training or trimming connectivity. However, CNNs take a different approach. At least one of the layers of a CNN is convolutional, meaning it passes an input matrix through a convolutional filter.

A machine learning algorithm would have to learn a distinct weight for each cell in a big tensor if it did not use convolutions. Convolutions allow a machine learning algorithm to decrease the amount of memory required to train the model, and build or extract patterns of increasing complexity using simpler designs marked by their filters. Convolutional filters are often seeded with random numbers in machine learning, and the network subsequently trains the ideal values [52].

Recurrent Neural Networks

Recurrent neural networks (RNN) have a very basic structure (see Fig. 2.10), which is however powerful for modelling sequential data, such as time series or natural language.

The main difference is that FNNs generate one output vector from one single input vector, whereas RNNs take a sequence of inputs. RNNs reuse prior input and output data. As shown in the figure, previous run's hidden layers supply part of the input to the same hidden layer in the following run.

Therefore, RNNs may consider longer sequences as elements, and are especially good for assessing them, because the hidden layers may have learned from prior runs on earlier portions of the sequence. That is, recurrent neural network training must be prolonged for each time step, which takes a long time and uses a lot of memory. RNNs are thus, in a sense, the deepest of all Neural Networks architectures. They are generic computers with higher processing capacity than FNNs, and they can generate and analyse memory of any sequences of input patterns [46].

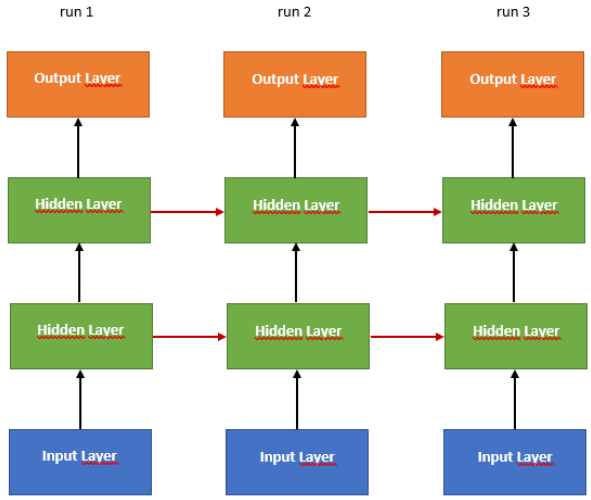


Figure 2.10.: Schematic structure of a recurrent neural network over time is shown, in which the hidden layers of the previous runs pass information to the hidden layers of the posterior runs.

Long Short-Termed Memory

Long short-termed memory neural network are a type of recurrent neural network that changed the field of speech and handwriting recognition, hence they are also used in machine translation and language modelling applications.

A cell, an input gate, an output gate, and a forget gate make up a typical LSTM architecture. The gates control the data flow into and out of the cell, and an internal memory state stores data across variable time periods. That way, LSTMs avoid the "vanishing gradient" problem that occurs while trying to train standard RNNs due to long data sequences. LSTM leads to a greater number of successful runs and learns considerably faster. LSTM also handles difficult, long-time-lag challenges that prior RNN algorithms have never been able to tackle [53].

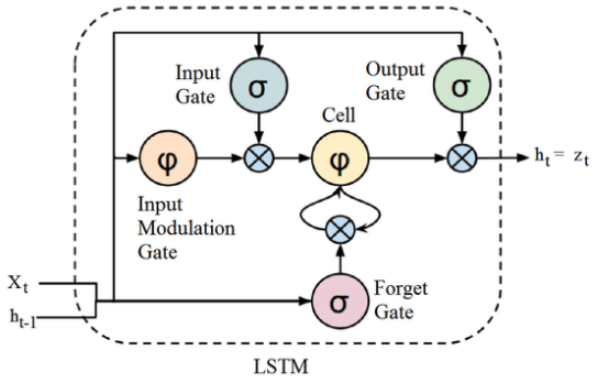


Figure 2.11.: Schematic structure of a long short-termed memory neural network is shown, in which the information flow through the input, output and forget gates is shown.
Source: [54]

2.3. Interpretability

The usage of machine learning (ML) systems is becoming more and more common, from self-driving automobiles and email classifiers to predictive police systems. They outperform people on certain tasks and frequently assist human thinking and decision-making processes. Therefore algorithmically informed decisions have an increasing influence on this computational society [55][56].

However, the majority of these precise decision-making models are still considered as sophisticated "black boxes", in which its internal processes are not revealed. These systems cannot be interpreted just by examining their parameters (e.g. a neural network). Especially, if we compare this with research in the development of new ML algorithms or models, research focused solely on the interpretability of these models remains a minority share. The aim of AI research has evolved away from the capacity to explain decision making and more towards building methods and algorithms that are centred on predictive capability [55][57].

Although ML algorithms appear to be effective in terms of forecasts, they are not without flaws. The most important is the opaqueness or lack of accountability, which is a feature of black-box ML models by definition [55]. Interpretability is frequently a big challenge, but it is needed to be taken into account alongside accuracy. The medical field is commonly used as an example of this need. Medical specialists must first comprehend the reasoning underlying the diagnosis, especially when the result is unexpected [5].

In order to solve this problem, Explainable Artificial Intelligence (XAI) is a new topic of study that concentrates on machine learning interpretability and aspires to create a more transparent AI. The major objective is to develop a set of easily understandable

models and methodologies that result in more understandable models while maintaining excellent prediction performance [55].

Interpretability does not have a mathematical definition. We define interpretability in the context of machine learning systems as the degree to which the reason for a judgement or a decision can be understood. The easier it is for someone to understand why particular predictions were made, the greater the degree of interpretability of the model [56][57].

Understandability, accuracy, and efficiency are often linked to interpretability. The first criterion is essential because as a definition, the model is interpretable if it can be understood. Also, in this context accuracy is required i.e., the hypothesis must have connection with the facts. And finally, efficiency, because every model could be understood in an endless period of time [5].

Although model size is typically linked to interpretability, Pazzani [58] noted that "no study has shown that people find smaller models more intelligible or that the size of a model is the sole element that impacts its comprehensibility."

To summarise, the simplest approach that machine learning may be made more comprehensible is by utilising interpretable linear models or decision trees, or by using "model-agnostic methods" that can be applied to any supervised ML model, which would be treated as a black box, even if it is not. The main advantage of this last strategy is that any machine learning model may be used, since sometimes the simplest models, although they are interpretable, are not accepted [57].

2.3.1. Importance of Interpretability

Choice auditing and verification have become mandatory as a result of new legislation and heavily regulated sectors, creating the demand for interpretable machine learning systems so that they can be examined, understood, and trusted [55]. The Royal Society, which is the Academy of Sciences of the United Kingdom (UK), released a study on its machine learning project in April 2017 [59]. The paper stresses interpretability and accountability, as well as responsibility and transparency, and considers them societal challenges linked with ML applications.

Also, in April 2018, the European Commission sent a message on Artificial Intelligence for Europe to a number of official European organisations, in which the necessity of research on the interpretability of AI systems is emphasised in order to further enhance people's faith in AI [60].

Not all machine learning systems need to be interpretable. In certain circumstances, it is sufficient to know that the predicted performance in a test dataset was good without knowing why a choice was reached. However, in medicine, criminal law, and other regulated sectors, there has been an emerging tendency to use machine learning for

high-stakes prediction applications that have a significant influence on human lives [55]. In this context, knowing the reason may help the expert to understand more about the problem and the possibility of failure of the model [57].

Some authors [56][55][57] agree that there are two types of scenarios in which interpretability and, as a result, explanations are not required: (1) when wrong findings have no substantial effect or severe repercussions, i.e., low-risk system (e.g. a movie recommender system); and (2) when the problem has been sufficiently explored and proven in real-world applications that we can trust the system's conclusions, even if the system is imperfect, i.e., well-studied system.

Interpretability is so crucial in the other cases because the forecast alone is just not sufficient for some problems or jobs. An accurate forecast not only partially answers the initial problem, but also the model must justify how it arrived at the prediction. The desire for interpretability and explanations is fuelled by the following factors:

Human curiosity and learning are satiated by interpretability. When something different than what people expected or are used to, an update of their mental picture is carried out by determining the cause of the unexpected incident. We do not require explanations for everything that occurs. Most people do not mind if they do not grasp how computers function. However, when researchers employ black-box ML models in their study, scientific discoveries are entirely buried if the model just makes predictions without providing any explanations. Interpretability and explanations are essential for learning and satisfying curiosity about why computers make particular guesses or behave in specific ways [55][57].

Find meaning in the world: This factor is related to the previous one and refers to the impulse to clarify discrepancies or conflicts between components of our knowledge. The answers do not have to entirely explain the problem, but they should address one of the major causes. Experimental evidence has shown that providing explanations increases acceptance of the interest recommendation [8]. The greater the impact of a computer's choice in people's life, the more critical it is that the system justifies its actions. That is why, as it was mentioned before, in criminal justice or, another example is, in loan granting, some sort of explanation is usually provided. Credit rejection must be legally justified by obvious grounds in some nations, which means the model justifying the denial must be interpretable [5]. Furthermore the need is urgent, since 2018 the candidate has the so-called right to be informed under the new European General Data Protection Regulation (GDPR), and a summary of all the deciding criteria may be requested [57][56].

Also, some ML models are **meant for science**, therefore, have the same goal: to gather information. Yet massive databases and black box algorithms are used. If the model is going to be the source of knowledge, interpretability is required to enable the extraction of this new knowledge [57].

Testing and safety procedures. Some systems are dangerous, e.g., an autonomous vehicle, and one needs to be very certain that the algorithms used works flawlessly [57].

In addition, a very important factor in favour of interpretability is that it is a really convenient **debugging tool** for identifying biases. Machine learning models, by nature, acquire biases from the training dataset. This could potentially make the algorithm discriminate underrepresented groups. For example, historically marginalised groups could be discriminated against in a loan-granting algorithm, implying not only that the company could lost potential profit, by losing customers, because the algorithm is not fully formulated. But also it is unethical, and the algorithm's creator should try to avoid discriminating against people based on their demographics [57].

Besides identifying biases, interpretability also allows for the discovery of erroneous model behaviour and anomalies. Since machine learning models can only be **debugged and audited** if they can be comprehended. Even in low-risk situations, such as movie suggestions. An explanation for a wrong forecast might assist you figure out what went wrong. It gives advice on how to improve the system [55][57][5].

To promote **societal acceptability**, incorporating robots and algorithms into our daily lives interpretability is needed. Usually, humans tend to assign robots or other animated objects with feeling, purposes, personal traits. And, in many cases, the only communication we have with the computer is an explanation, which the user may interpret as a social interaction between the two. As a result, an algorithm that justifies its results will be accepted more easily [57].

Strongly related with the previous factor, the explanations provided by the model are also **managed as a social interaction** by the programmer in order to have an impact on the behaviour, feelings, and opinions of the person receiving the explanation. Frequently, machines must convince humans for the purpose of accomplishing their intended aim [57].

These auxiliary conditions must be met for ML systems to be utilised securely. However, unlike performance metrics like accuracy, these criteria are not always totally quantifiable. Nevertheless, if the machine learning model results' can be understood, one can more easily evaluate the features listed below [56]:

- Fairness: Guaranteeing that forecasts are somehow fair and not biased against marginalised populations, either indirectly or directly.
- Confidentiality or privacy: Sensitive information must be kept private.
- Robustness and reliability: good performance levels can be assured because minor variations in the input will not result in big variations in the forecast.
- Causality: Ability to ensure that only causal connections are detected by the model. Meaning, that a perturbation's expected output change will also occur in the real system.

- Trust: it is more likely for humans to trust a system that explains its judgements than a "black box" model.

Specifically, interpretability is also important for human-centred production. The presence of this concept in the industry is obvious. In the recommended and responsible practices for AI published by Google [61], they defend the importance of interpretability and one of the recommended practices is to treat interpretability as a fundamental part of the user experience.

2.3.2. Fundamentals of XAI

As it was already mentioned previously, as a result of the growing adoption of progressively more complicated models based on deep learning approaches (e.g., face recognition, speech to text, etc.), Explainable AI (XAI) has arisen as a new topic of study in machine learning that tries to solve how "black box" judgements of AI systems are made [55][62]. Furthermore, the European Union (EU) is working to formalise the definition of "trustworthy AI," with transparency being one of seven important elements. With the adoption of GDPR and the definition of the same degree of safeguards for its citizens with AI, the EU set the standard [63].

Explainable Artificial Intelligence (XAI) refers to methodologies and strategies used in the application of artificial intelligence (AI) technology that allow skilled people to understand why AI systems reach the solutions they do [64]. This discipline examines and attempts to comprehend the methodology that go into the decision making. Most owners, operators, and consumers expect XAI to provide answers to several key questions. However, using a XAI method to AI model development might raise the initial investment required to meet model transparency standards [62][63].

The concept of explainable artificial intelligence is closely related to that of interpretability. Some authors [55] use the terms XAI, interpretable AI or interpretable machine learning (ML) interchangeably, as so will I during this thesis. While other authors consider that a model is interpretable when it can be understood by a human without external help, simply by looking at the parameters of the model, such as linear models or decision trees. Explainable models, on the other hand, are too complex to comprehend (e.g., neural networks) and need the use of extra tools (such as model-agnostic methods) in order to explain how they produce predictions [65].

XAI is at the heart of human-centric software systems such as recommender and decision-support systems for e-Learning and e-Health. Moreover, building conversational robots capable of providing humans with coherent, compelling, reliable, and successful interactive explanations is one of the key problems of XAI [66].

The interpretability problem refers to the technical challenge of explaining AI choices. Another factor to consider is information overload (also called sometimes infobesity),

thus complete openness may not always be attainable or even needed. However, simplification at the risk of deceiving users to improve trust or hiding unwanted system characteristics should be avoided by permitting an equilibrium between interpretability and fidelity of an explanation [67].

There's a rising argument for explainable AI, but what should XAI be attempting to explain? The National Institute of Standards and Technology (NIST) [68] in the United States has established four principles within the explainable AI framework. These principles are a collection of rules for the essential qualities that explainable AI systems should possess [64].

1. Explanation: For each output, this principle requires AI systems to provide proof, support, or rationale. However, this rule does not require that the evidence is right, instructive, or understandable. This idea imposes that all AI systems' outputs should be accompanied with evidence or justifications without quality criterion on their explanations because the following two principles will be used for assessing the quality of the explanation.

2. Meaningful: The system meets this criterion if the explanations are intelligible to particular users and/or they are beneficial to finish a job. This does not mean that the system should give only one good explanation useful in every situation for every user. Multiple adapted explanations may be required for distinct groups of users (e.g., developers vs. end-users).

3. Accuracy: This principle demands that the information given accurately explains how a system generates outcomes. Decision accuracy is not the same as explanation accuracy. When it comes to decision tasks, the first one relates to whether or not the system's judgement is right. However, how the system arrived at its conclusion may not be adequately reflected in the associated explanation.

   Although validated decision accuracy measurements exist, academics are now working on performance measures for explanation correctness [68]. Again, multiple explanation accuracy measures for different groups and people are possible. Some users will demand concise descriptions that focus on the essential points, whilst specialists may require the details to fully evaluate the system's output generation procedure.

4. Knowledge Limits: Systems should only be used in the conditions that they were designed for. This criterion declares that algorithms should detect situations in which they were not intended to function, or in which their responses are not trustworthy, based on an internal confidence level. This method protects responses by identifying and disclosing knowledge boundaries, ensuring that no judgement is made when it is not necessary. By prohibiting deceptive, hazardous, or unfair decisions or outputs, the Knowledge Limits Principle can build confidence in a system.

3. State of the Art

In this chapter we survey the literature on inertial sensors and the models currently used for Human Activity Recognition, interpretability, and privacy regarding data collection. The empirical studies that influenced our research are going to be described. And also, the current practices, limits and criticisms that have been raised in the literature will be explored.

3.1. Activity Recognition with Smartphones

Human activity recognition research has become a frequent topic at major international conferences [42]. Nevertheless, the literature on this subject is more developed in applications aimed at the healthcare environment and ambient assisted living. While worker activity recognition in the industrial industry is still a recent topic, with just a few investigations to date [26].

The idea behind human activity recognition (HAR) is that body motions can be translated into distinct data patterns that may be detected and categorised using machine learning algorithms [52]. However, some systems attempt to detect user's activities merely based on their location. This type of systems is based on utilising geolocation sensors and algorithms such as Hidden Markov Model or Support Vector Machine [69].

The first sort of HAR, in which the sensors identify behaviour from movement, is the one that has received the most recent attention [69]. This form of activity recognition is divided into two categories in terms of the activity monitoring sensor: activity recognition based on vision vs. activity recognition based on sensors [42].

Because of its importance in fields like surveillance, robot learning, and anti-terrorist security, vision-based activity identification has been a study of interest for a long time. Researchers have investigated a range of applications for a single user or groups of humans using a variety of modalities, such as a single or multicamera settings, stereo, and infrared cameras [42].

There is a dearth of comprehensive reviews on the state of the art in sensor-based activity recognition when compared to the number of surveys in vision-based activity [42]. This might be because the technique was only recently made possible when sensing technologies advanced to the point where they could be deployed in a realistic manner in terms of communication infrastructure, prices, and sizes. Current sensors' availability has enabled the development of a wide range of practical applications in a variety of fields, including health, the Internet of Things and Smart Cities, security, and transportation

[70]. Moreover, sensor-based activity recognition, in particular, has the potential to handle profound concerns such as privacy, ethics, and obtrusiveness more effectively than traditional vision-based techniques. That is one of the reasons why, for this project, in which the interpretability and privacy of data serve as one of the main motivations, specifically mobile-phone-collected data, will be used.

A very similar division was proposed by Rasnayaka and Sim [71]. These authors propose a HAR approach through gait. Gait is defined as a person's ambulation pattern, which includes walking, running, and climbing stairs. They divide gait into two areas: visual gait, the area that uses external cameras as sensors, and the discipline that uses wearable IMU sensors, called on-body gait.

The concept of employing sensors to monitor and recognise activities has been around since the late 1990s [42]. The work of M. Mozer in 1998 [72] in the context of home automation was the first to pioneer and experiment with sensorics for HAR. After that, wide research has been carried out to investigate sensor usage in various ubiquitous and mobile computing application situations. This leads to significant work on smart appliances, context awareness, and activity detection [42].

Wearable or portable sensors were employed in the majority of studies at the time. In these studies, physical activities like standing, walking, and running were monitored. Nevertheless, a novel sensor-based strategy of monitoring human behaviours arose in the early 2000s, using sensors connected to objects. This methodology, later named "dense sensing," accomplishes activity detection via user–object interactions. The technique is especially well suited to dealing with activities that involve a large number of items in a given environment, as well as instrumental activities.

Much research has been conducted and great progress has been achieved in wearable sensor-based and dense sensing-based activity recognition, however the two primary techniques to activity detection are still being studied. The first is mostly propelled by ubiquitous and mobile computing, whereas the latter is primarily pushed by smart environment applications like Ambient Assisted Living (AAL) [42].

Sensorics for HAR

In this work we are going to focus mainly on the study of wearable-sensor based activity recognition, as this allows activity and context recognition regardless of the location of the user. This area of research has been greatly affected by the recent availability of inertial measurements units (IMUs) on smartphones, smart bands, and smart watches to monitor hand movement. Wearable (on-body) sensing is based on combinations of sensors. A set of tri-axial accelerometers, gyroscopes, and magnetic field sensors are included in each IMU [73]. The accelerometer and gyroscope are the most popular inertial sensors used to gather information about the human body's acceleration and direction of motion, respectively. These sensors have enabled the gathering of a wide range of data about the user, which may be used to identify certain physical activities [70].

Specifically, because of their widespread availability, smartphones are frequently used to create HAR solutions. Mobile phones are one of the most often utilized instruments for identifying human actions because of its portability and feature processing capacity, networking capabilities, and the range of integrated sensors [70].

Lane et al. [74] suggest four reasons why a smartphone is an incredible tool for identifying human actions. To begin with, a smartphone is a low-cost gadget that combines multiple software and hardware sensors into a single device. Second, they are programmable and open devices. Third, through delivering information and applications via virtual marketplaces, smartphones have a large mass reach. Finally, cloud computing enables developers to add more functionality to this equipment that act as support and information sharing. In conclusion, the ability of smartphones to (1) gather and analyse data, (2) transmit and receive data, and (3) link with other devices or sensors in the physical world provides a significant advantage over other wearable devices [70].

One of the first historical milestones that characterized the evolution of the HAR field from the standpoint of smartphones was in 2006 [75], when the first HAR solutions that expressly employed mobile phones arrived. At the time, the first investigations were conducted utilizing data analysis gathered from Global System for Mobile communication (GSM) sensors and accelerometers to track users' movements. Because mobile phones had limited computational capacity, all data processing was done on a computer (offline processing) during this time [70]. The literature would later progress to the creation of the first joint solutions [76].

Later methods performed data gathering and analysis on the smartphone itself, and advances in mobile phone sensors allowed for the recognition of new activities. Many researches were focused on the creation of applications in the healthcare field at the time, such as chronic illness identification based on the users' locomotor concerns [70]. The first experiments in recognizing more sophisticated behaviours with smartphones were done in 2012. For example, Dernbach et al. [77] used data from inertial sensors to identify everyday (e.g. cooking) and other physical activities.

Data Processing for HAR

HAR systems based on smartphones with inertial sensors have grown and followed a development approach with well-defined phases such as collection of data, segmentation and merging, feature extraction and selection, and machine learning algorithms to generate classification models [70]. Banos et al. in 2014 focused on the data segmentation stage, with the goal of determining the influence of the size of the time window on classification model accuracy [78].

Due to the flexibility of hand motions and the appearance of sensor noise, data collected with wearable sensors comprises of extraordinarily complex contexts of signal fluctuations. Therefore, in general, most classifiers are unable to accept raw sensor data as input. For most applications, extracting and choosing features to discover useful context information for categorization is required. By categorising IMU signal features, one

may distinguish hand actions using unique classification techniques [73]. According to W. Tao et al. [26], the problem of activity recognition may be divided into two parts: feature extraction and subsequent multiclass categorization.

The majority of recognition algorithms choose characteristics from a set of "engineered" features [52]. In 2020, A. Kempa-Liehr et al. quantify each time-series in terms of its distribution of values, correlation characteristics, stationarity, entropy, and nonlinear time-series analysis using the FRESH algorithm (FeatuRe Extraction on the basis of Scalable Hypothesis testing). To avoid overfitting, this brute force feature extraction is computationally intensive and must be followed by an identification of feature relevance [79]. The feature selection technique is also time-consuming and leads to a complexity in "scaling up" activity identification to sophisticated high-level behaviours [52].

D. Figo et al. [80] present methods for extracting activity information from accelerometer data in raw form. These techniques rely on translating or manipulating input signals across distinct representational domains. The time domain, frequency domain, and what they term discrete representation domains are the key domains in which it is feasible to classify the various sensor signal processing algorithms, as shown in Figure 3.1. In order to assess their implementation complexity and accuracy in extracting signal characteristics and recognising user behaviours, the following subsections discuss the most typical approaches in each of these categories.

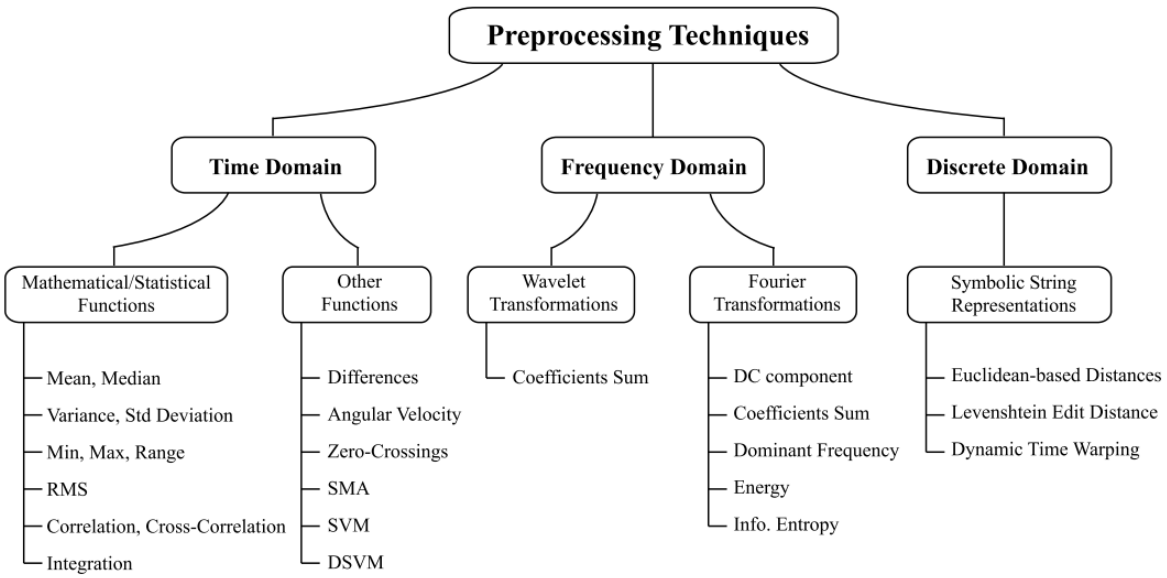


Figure 3.1.: Figo et al. proposed techniques for extracting features from sensor signals for activity classification.
Source: [80]

Recognising Activities

According to L. Chen et al. [42], there are two basic types of data-driven activity modelling: generative and discriminative. Generative activity modelling tries to build a complete description of the input or data space, typically using a probabilistic model like a Bayesian network, and discriminative activity type only models the modelling from inputs (data) to outputs (activities). This research is going to focus on the second approach, which is used more commonly.

Support Vector Machine, Random Forest, and K-Nearest Neighbors are some of the classifiers that have been investigated for activity recognition with the features [26]. L. Chen et al. [42] also states that the simplest discriminative strategy is probably nearest neighbour (NN). Bao and Intille [81] studied this technique for the identification of activities using accelerometer data, as well as a number of other classification models. They discovered that decision trees outperform the basic NN technique. Furthermore, while the decision tree technique has the advantage of producing rules understandable by the user, and it is thus easily interpretable, it is prone to brittleness when dealing with high-precision numeric data [42]. Support Vector Machines (SVMs) functioned reliably well according to Ravi et al. [82]. They discovered that when recognising a collection of eight activities, a simple method functioned best for three easiest categories, whereas their SVM model performed best for the most challenging one. D. Anguita et al. [83] used a variation of the SVM algorithm to achieve improvements in recognition accuracy and battery consumption in healthcare HAR applications that also used smartphones.

In this context, deep learning algorithms provided a new way to deal with increasingly complicated data. Artificial neural networks are powerful for feature extraction and are at the base of deep learning methods. Raw sensor inputs can be fed into neural networks, whether recurrent or feedforward. It has been recommended that convolutional networks (CNNs) be used to extract features from raw sensor information in order to improve performance. CNNs have shown good performance in feature extraction from time series using a signal convolution operation with a filter (or kernel) [52] [84]. CNNs have become the state of the art deep learning technique for HAR [73].

Typically, in the wearable HAR domain, neural network topologies that integrate convolutional and other types of layers are implemented [52]. Many traditional approaches have been outperformed by deep learning algorithms. With the implementation of deep learning classification methods, the HAR area began to converge in 2015 [70]. According to F. Ordoñez and D. Roggen [52], deep neural networks may perform more sophisticated input transformations than shallow networks i.e., networks with a smaller number of hidden layers or a single hidden layer.

In the literature, there are a variety of different architectures used for this application after the feature extraction. For example, an input layer, an output layer, and a single hidden layer of eight units make up the construction of the ANN utilised in Anderson and Muller's work [85]. The ANN is supplied with two features based on the authors' observations. However, it is more common that raw signals, which are obtained from

wearable sensors, are then processed by convolutional networks combined with dense layers to generate a probability distribution over various human activities. These network topologies outperformed state-of-the-art techniques in terms of discriminative power, according F. Ordoñez and D. Roggen [52].

Using recurrent neural networks (RNNs), a sequential modeling strategy, has recently been used for time series domains with positive results [73] [52]. The RNN method allows the user to take into consideration both current and prior input data. On activities that are brief in length but have a natural ordering, recurrent networks outperform other types of networks considerably [84]. Specifically, researchers P. Rivera et al. [73] and F. Ordoñez and D. Roggen [52] claim that framework for activity detection based on long-short-term memory (LSTM) networks are appropriate for multimodal wearable sensors. LSTMs are recurrent networks with a memory to describe temporal reliances in time series applications. In the voice recognition area, where modeling temporal information is necessary, the coupling of CNNs and LSTMs in a unified framework has already provided state-of-the-art results [52]. The LSTM memory unit enhances the abstraction of sequential input data, such as HAR, because they are well suited to learning temporal dynamics in sensor signals.

While deep models have been explored for a range of situations in HAR, there is yet to be a thorough examination of deep learning capabilities. The authors claim to have performed exploratory experiments to investigate the parameter space, but they frequently leave out the specifics. For instance, the global feature extraction procedure is still a mystery and difficult to duplicate. There are questions that remain unanswered such as: How probable is it for the next person to obtain the parameter setting that performs equally well for their application? Or, which features have the greatest influence on the performance? [84]

Also, while some research has used CNNs to recognise activities, the successful combination of convolutional and recurrent layers, which has previously produced state-of-the-art results in other time-series domains like speech recognition, has yet to be extensively researched in the HAR domain [52].

3.2. Challenges for Activity Recognition

In addition to the state of the art in the recognition of human activities and the sensors used for such applications, in this section, the state of the art of interpretable machine learning models will be described too. Regarding this topic, while scholars' interest in model interpretability has expanded fast in fields such as HCI, ML, etc., little is known about how researchers view and try to deliver interpretability. Motivated by the recent growth of ML's application area, the concern of the human capacity to understand its functioning and its results has also increased. This gets added to the fact that the areas to which these models are applied are increasingly sensitive, affected by issues such as data privacy, or in which errors can cause fatal consequences [86].

However, some studies show a more complex picture of interpretability, compared to the optimistic image shown by others about interpretability as a tool. For example, according to Bussone et al., users of clinical decision-support models tend to over-rely on the model's advice over their own knowledge [87]. Narayanan et al.'s findings show that explanation complexity affects negatively to efficiency and user approval but not certainly on accuracy [88]. The amount of input features and model transparency impact the user's ability forecast model behaviour and trust, according to Forough et al. [89].

Furthermore, Ray Hong [86] points out that there are few studies trying to understand how ML experts conduct interpretability-related jobs and what their methodology, requirements, and problems are. Although interpretability is frequently defined as how effectively a model conveys its conclusions to a user, little is known about how interpretability emerges in real-world workspaces where teams must interact and coordinate their efforts around models and decision-making tools.

In other words, these lines of study imply that model interpretability and its impact are complicated, and that they may be influenced by consumer characteristics and circumstances. Therefore, a more exhaustive study on the subject is required. Other authors, for example, Doshi-Velez and Kim emphasize the importance of model interpretability. They consider it an indicator, not only for task performance, but also for auxiliary requirements, including safety, privacy, non-discrimination, justice, avoiding technological debt, dependability, offering the right to explanation, trust, and more [56].

Another important challenge that affects activity recognition is data privacy. Personal electronics such as smartphones and wearable devices such as smartwatches, fitness trackers, etc. have exploded in popularity recently. Because of their low cost and minimal demand on power and memory. Almost all of these devices have inertial measurement units (IMU). As these gadgets become more widely used, more data is collected as people wear or carry them in their daily lives. Therefore, many fascinating applications, including activity detection, health monitoring, step tracking, gait-based authentication, and continuous authentication, have been jeopardised as a result of this data [71].

There are many studies on how to predict gender, age, and other physical characteristics, such as height, weight, BMI using data provided by sensors such as facial images, fingerprints, iris, or movement with IMU sensors. Rasnayaka and Sim [71] study which additional user details, including height, weight, BMI, age, gender, and activeness, can be accurately learnt from the data collected by the wearable sensors, apart from the activity currently performed, which would be the intended use of the application. There can be several unanticipated privacy concerns because, in the on-body gait discipline, one or multiple devices with an IMU sensor must be linked to the subject's body at all times. They examine physical, socioeconomic, and psychological variables to determine the extent to which gait might provide information.

Not all characteristics are equally essential; for example, exposing one's weight may be more intrusive on one's privacy than revealing one's gender. Because the relevance

or sensitivity of the projected personal attribute is subjective and relative to each user, Rasnayaka and Sim [71] look at the relative relevance of each attribute to compute a Privacy Vulnerability Index (PVI). The PVI is a value that includes, on the one hand, the precision with which it is able to predict a specific characteristic, and on the other, the relative importance that users give to this attribute based on a survey.

3.3. Achieving Interpretability for HAR

The literature distinguishes between transparent models, which are interpretable by design, and models that can be explained using external XAI techniques, often known as the post-hoc explainability approach. This criterion differentiates whether interpretability is accomplished by limiting the complexity of the machine learning model (intrinsic) or by using post-training analysis tools (post hoc). Transparent models, such as brief decision trees or sparse linear models, are deemed interpretable due to their basic structure. Visual explanations, text explanations, local explanations, explanations by simplification, explanations by example, and feature relevance explanations approaches are all used in post-hoc explainability to target models that are not easily interpretable by design [57][90].

- **Visual explanation** approaches aim to visualize the behaviour of the model. Many techniques use dimensionality reduction techniques that enable interpretable simple visualization
- **Text explanations** learn to generate text description or symbols that depict the model's operation.
- **Local explanations** provide explanations to less complicated but important solution subspaces by segmenting the solution space.
- **Explanations by simplification** methods involve rebuilding a totally new system based on the taught model to be described. The new model tries to maintain a comparable performance score while has lower complexity.
- **Explanations by example** extract representative examples that cover the core correlations found by the model being analysed, similar to how humans operate when attempting to explain a given process.
- **Feature relevance explanation** approaches compute a relevance score for the variables to elucidate the inner workings of a model. These scores indicate how much of an impact a feature has when generating the output.

Furthermore, a distinction is made between model specific and model-agnostic interpretation tools. Unlike the other type, model-agnostic methods can be used on any ML

model. These tools are often used after the model has been trained (post hoc) and work by analysing feature input and output pairs [57].

Although performance evaluation on validation dataset is a valuable approach for any application, it is possible that it does not reflect performance on additional "real-world" data and thus confidence cannot rely exclusively on it. Examining examples provides an alternate technique for determining the model's reliability, especially if the examples are explained. As a result, Ribeiro et al. [91] propose explaining numerous illustrative individual model predictions as a method to convey a global perspective.

It is crucial to distinguish between trusting an individual prediction enough to act on it and trusting a model to behave in acceptable ways if utilised. In their study, Ribeiro et al. [91] suggest that a solution to the "trusting an individual prediction" may be individual explanations to them. And a solution to the "trusting the model" problem might be choosing several individual predictions to explain.

Therefore, they recommend **LIME** (Local Interpretable Model-agnostic Explanation), a black-box approach that can explain any classifier or regressor's predictions accurately by approximating it locally with an interpretable model. In combination with SP-LIME, an approach that picks a collection of representative cases with explanations to address the "trusting the model" problem. In the conducted experiment, non-experts using LIME are able to pick which classifier generalizes better in the real world.

The basic purpose of LIME is to find an interpretable model that is locally loyal to the classifier across the interpretable representation. To do this, the algorithm investigates predictions when different datasets are fed into the ML model. Firstly, LIME creates a new dataset containing perturbed samples from an individual observation and their corresponding predictions from the black-box model. The sampled instances are weighted by the distance closeness between the generated data and the original instance.

The algorithm then trains an interpretable model on this new dataset, experimenting with different combinations to find the K features that best described the complex model outcome from the permuted data. Any interpretable model, such as linear model, decision trees, or falling rule lists, can be used. The interpretable model's feature weights are used to explain the chosen individual observation. The learnt model should be a good local approximation of the ML model prediction, but not necessarily a good global approximation.

Mathematically, the LIME algorithm for obtaining the explanation is as follows [91]:

$$\xi(x) = \min_{g \in G}[L(f, g, \pi_x) + \Omega(g)] \tag{3.1}$$

The explanation, ξ, for instance x is the model g, which is the interpretable model that minimizes the loss function, L, with respect to the original function, f, while model complexity $\Omega(g)$ also remains low. π_x is the proximity measure between the instances and x.

As a result of LIME algorithm's application, the user gains some comprehension and trust in the classifier as a result of the explanation of a single prediction. This, however, is insufficient to assess the model's completeness and give it confidence. Some researchers employ RandomPick-LIME (RP-LIME) algorithm to provide a global explanation of the model. RP-LIME's approach entails explaining one random LIME instance to explain the model's behaviour locally. However because a local explanation does not provide us a complete picture of the models functioning, SubmodularPick-LIME is another option, which has been proved to outperform RP-LIME with uniformed users [91]. Therefore, the usage of the Submodular-Pick LIME (SP-LIME) method is recommended to gain a global comprehension of the model explaining only certain particular predictions.

SP-LIME remains model-independent. It entails the careful selection of a small number of forecasts so that users do not have to study at a vast number of explanations. The budget, B, refers to the number of explanations that the user is ready to investigate in order to comprehend the model, i.e., the number of different and representative explanations that the algorithm must find. Furthermore, SubmodularPick allows us to produce candidate explanations from the complete database or a random sample of the required size to minimize computational time.

The pick step is described as the duty of selecting the set of B instances that the user will inspect from a collection of predictions or a random sample of them. During this step, an importance value is assigned to each of the input features that the model uses to create the prediction. Intuitively, the model would provide higher importance value to features that are able to explain a bigger number of diverse predictions with more weight.

As a result, when selecting the most representative predictions, we will want to select the forecasts whose explanations incorporate the bigger number of significant components. Nevertheless, we will also want these explanations not to be redundant in their significant features, that is, we will want to avoid picking forecasts with matching explanations.

Recently, saliency maps have become increasingly popular. They are a visualization tool for understanding why a deep learning model made a certain choice, e.g., when categorizing an image. Simonyan et al. [92] use **Saliency Maps** to understand deep learning classification models. Saliency maps capture the most unique feature in an input. These tools are a topographic representation of the most unique inputs, e.g., pixels, since usually these explanation algorithms are used in vision-based recognition techniques.

The objective of the algorithm is to rank the input elements, by the example of an image I, the pixels of this image, based on their influence on a classification algorithm using the class score function $S_c(I)$. This approach is well-known in the field of computer vision, but it may also be used to explain deep time series classifiers [93]. Being the linear score model for the class c:

$$S_c(I) = w_c^T I + b_c \tag{3.2}$$

where I is the 1-D image vector, and w_c and b_c are the weight vector and the bias vector respectively. The influence of the respective pixels of I for the class c is determined by the value of the elements in w.

However, the class score function $S_c(I)$ is a highly non-linear function of I for complex DL algorithms. As a result, the preceding method cannot be applied right away. Nevertheless, given an image I_0, using the first-order Taylor expansion, $S_c(I)$ may be approximated with a linear function in the vicinity of I_0:

$$S_c(I) = \frac{\partial S_c}{\partial I}\bigg|_{I_0}^T I + b_c \tag{3.3}$$

The magnitude of the derivative suggests which input elements need to be modified the least to impact the class score the most, according to Simonyan et al.'s interpretation of estimating image-specific class saliency using the class score derivative [92].

Schreiber [94] also recommends an algorithm called Vanilla Gradient to generate Saliency Maps. Vanilla Gradient has demonstrated to be quite robust, and also the simplest method among gradient-based techniques. In summary, the way this algorithm works is firstly, a forward pass is made with the data. Then the backpropagation is performed, which would normally be done during training only up to the second layer since the input cannot be changed. Vanilla Gradient, on the other hand, continues the backpropagation to the input layer to check which pixels have the greatest impact on the outcome. This is the reason for the simplicity of this algorithm. After this step, the gradient would be rendered as a normalized heatmap.

Petsiuk et al. [95] propose a different approach applicable to images called **RISE** (Randomized Input Sampling for Explanation of Black-Models) that generates an importance map showing how important each pixel is for the final prediction. As its name indicates, RISE is a more general approach that works with black-box models, as opposed to white-box techniques that use gradients or other internal network information to assess pixel significance.

The main objective of RISE is to measure the significance of an image area. To do that, the algorithm perturbs the input image with blur, noise, or changing random intensities to zero to randomly mask the picture. The model is supplied with these additional inputs once the random masking is completed, and the precision value with which the outcome is identified in these new cases is kept.

As a result, the final saliency map may be calculated as a weighted sum of the randomly generated masks. The weights of the linear combination are the probability scores provided from each mask, adjusted for the random masks' distribution.

Lundberg and Lee [96] published an algorithm named **SHAP**, which is the abbreviation for Shapley Additive exPlanations. SHAP's publication was motivated by the need to anticipate the risk of intraoperative hypoxemia, and to offer an explanation of the features that contribute to that risk when under general anaesthesia [97].

SHAP uses combination of feature contributions and game theory. SHAP values are derived from Shapley values, a game theory notion. Shapley values are a way of allocating rewards to participants based on their contribution to the overall payment. Players form a coalition in order to get a specific benefit from their collaboration. The "players" in this example are the individual features cooperating in the model to obtain the "benefit", which corresponds in this case to the actual prediction minus the average prediction. The "game" is obtaining the model's prediction for an instance [98].

More technically, SHAP assigns an importance value to each feature that represents the effect on the model prediction of including that feature. SHAP values may be used to explain individual predictions. Since, SHAP values quantify the influence of each characteristic on the prediction for a given instance. To compute this effect, assuming independence of input features, a model is trained with that feature present, and another model is trained with the feature withheld. The same process is repeated, training the predictive model for each distinct feature coalition, meaning $2^{features}$ models. Then, the predictions from every model are compared on the same input observation (x_0).

SHAP values are computed using the marginal contribution. The marginal contribution brought by feature i to the model containing only i as feature is:

$$MC_{i,\{i\}}(x_0) = predict_{\{i\}}(x_0) - predict_{\emptyset} \tag{3.4}$$

However, to obtain the overall effect of feature i on the final model it is necessary to consider the marginal contribution of i in all models where that feature is present. The Shapley values are then computed and used as feature attributions. Shapley values are the weighted average marginal contribution of a feature of all possible coalitions. Concisely, given a f-featured model, the SHAP value of i is:

$$SHAP_i(x_0) = \sum_{set:i \in set} [|set| \times \binom{f}{|set|}]^{-1} \times MC_{i,|set|}(x_0) \tag{3.5}$$

where set are all the possible coalitions of features, $set : i \in set$, corresponds to all the possible coalitions of features of which i is part of, and $|set|$ is the number of features that each of these coalitions has [98].

3.4. Summary of State of the Art

To conclude, the main techniques on the recognition of human activities are based on data provided by inertial or visual sensors. With regard to inertial sensors, the use of IMU-collected data is very frequent since IMUs are present even in smartphones. IMUs

mainly consist of an accelerometer and a gyroscope. However, with these sensorics, very sensitive information about the user can be obtained, which raises concerns related to data privacy. For this reason, one of the objectives of this work is closely related to the interpretability of the machine learning models used in HAR applications. Besides, as has already been explained, some authors in the literature are wary of exclusively attending to accuracy measures for a model's evaluation. They recommend the complementary utilization of algorithms able to provide an explanation about the model's behaviour, such as LIME, SHAP, RISE or Saliency Maps.

The algorithms frequently used in HAR applications range from k-nearest neighbour, decision trees to ANN. Lately, neural networks composed of convolutional networks that can extract features from the raw data collected by sensors are becoming more popular. Besides the aforementioned models, algorithms can be composed of recurrent neural layers, specifically LSTMs, due to the temporary information obtained in these applications.

4. Development and Evaluation

4.1. Human Activities and Postural Transitions Dataset

Human Activities and Postural Transitions Dataset (UCI HAPT) dataset [88] is made up of a collection of complex naturalistic behaviours that were recorded in a sensor-rich environment. It features recordings of a group of thirty volunteers, within an age bracket of 19-48 years old, engaging in a protocol of activities composed of six basic activities both static (standing, sitting, and laying) and dynamic (walking, walking upstairs, and walking downstairs). Postural transitions between the static postures, e.g., stand-to-sit, sit-to-lie, stand-to-lie, etc., were also included in the trial.

This database is an updated version of UCI Human Activity Recognition Using Smartphones database, and it is available on the following repository [99]. Instead of the pre-processed inertial signals from the smartphone sensors that were supplied in the previous version, the current one also offers the original raw inertial smartphone sensor signals. This modification was made so that activity recognition could also be conducted using raw data. Furthermore, activity labels were changed to incorporate postural changes that were not included in the prior dataset. This database has been used by numerous third-party publications (e.g., [100] [23]).

The training-test data split was previously performed randomly by the authors of the dataset. The data of the sessions form twenty-one subjects (70%) was selected for generating the training data with which our model was trained. Then, we report classification performance on a testing composed of the remaining seven (30%) subjects, which corresponds to the test data.

In terms of sensor setting, during the experiment, each participant wore a smartphone (Samsung Galaxy S II) around their waist. Using the device's built-in accelerometer and gyroscope, 3-axial linear acceleration and 3-axial angular velocity were recorded at a constant rate of 50Hz. These tests were videotaped in order to manually label the data afterwards. The axis orientation of the smartphone's accelerometer is shown in Fig. 1.

Figure 4.1.: Image showing Samsung Galaxy S2, smartphone used by Reyes-Ortiz et al. to capture data. Arrows show the axis orientation of the accelerometer. **Source:** [88]

There are two types of data in this database, each of which will be utilized independently:

1. Raw data from inertial sensors (accelerometer and gyroscope) and a list of all the actions that have been completed. This data is going to be fed to neural network in this study, since convolutional layers can extract features from it.

2. Activity window records, each of one includes a 561-feature vector containing variables in the time and frequency domains, the label for the related activity and a unique identity for the person who conducted the experiment. The feature vector is going to be used as input for the classical ML algorithms studied in this work.

The sensor data (accelerometer and gyroscope) were pre-processed using noise filters before being sampled in 2.56-second-fixed-width sliding windows with 50% overlap (128 readings/window). A Butterworth low-pass filter was used to separate the gravitational and body motion components of the sensor acceleration data into body acceleration and gravity. Because it is expected that the gravitational force has only low frequency components, a filter with a cut-off frequency of 0.3 Hz was utilized. Calculating variables from the time and frequency domain yielded a vector of 561 features from each frame.

4.2. Data preparation and Evaluation approach

According to the authors of the UCI HAPT database [88], the accelerometer is the most commonly used sensor for reading body motion signals. Since the objective of the practical part of this study is not only to obtain the algorithm that works best. Rather, it seeks to obtain a compromise between the precision and the interpretability of the model. There are already several investigations about the best models with greater accuracy for HAR applications, as it has already been presented in the chapter on the state of the art, and yet the issue of interpretability has not been studied as much. Consequently,

and regarding data privacy fewer user information will be used, considering only the raw data obtained from the accelerometer and the gyroscope as neural network input. The total acceleration measured by the sensor will not be used, as the gravitational component is not of interest. Thus, only body acceleration is used.

Since 561 features are possibly too many to achieve a good model interpretability, only those features corresponding to the time domain will be fed to the classical ML algorithms. We argue that frequency domain features, while explainable, are not as easily interpretable. Also, not even all time-domain features are going to be used. Only the features respective to the mean, max and min values of each feature will be used as input. Thus, parameters that have been considered to be less interpretable have been excluded such as the standard deviation (std), the autoregression coefficient (arCoeff) or the correlation coefficient (correlation). This consideration has been made because it is to be believed that the worker needs a higher level knowledge of statistics to interpret correlation compared to interpreting mean or min/max values.

As a result, instead of the 561 features previously described, only 60 features were fed to the machine learning models. Usually, the amount of input data fed to the model and its resultant accuracy are strongly linked, as long as it is not redundant [101]. However, having only 60 characteristics helps us to obtain a better understanding of the model. One of the goals aimed to be obtained with the practical part of this thesis is to evaluate the trade-off between accuracy and interpretability. Accuracy and the model's performance cannot be used as the sole evaluator because of the necessity for transparency and interpretability to discover patterns, biases, and errors, as well as the rising concern and regulation on data privacy and cybersecurity.

Prior to being fed to the models both the raw data and the features of this research were subject to preprocessing. After loading the data into the model and selecting just those that will be used to train and evaluate our network based on what was previously stated, it is required to normalize the input data by subtracting the mean and scaling it to unit variance. In addition, the output data utilized in the models has been submitted to one-hot encoding, excluding the data fed to the decision tree model, for which this is not necessary.

Several traditional machine learning classification methods and also neural networks will be implemented in this project. These models will be compared in terms of their performance and their capacity to be explained by interpretability algorithms. The classical ML algorithms that are going to be employed are logistic regression, decision trees and k-nearest neighbour. A deep learning approach will also be implemented and compared with the three previous classical approaches.

The models have been implemented once the features that are the most interpretable have been chosen. The flowchart in figure 2 depicts the overall methodology process that has been established. First, the model and the hyperparameters that describe it will be broadly defined. The hyperparameters for which the model achieves the best results

for this database will then be picked using the GridSearchCV or RandomizedSearchCV functions. Finally, when the model has been trained, it will be assessed using database instances that it has never seen before.

Figure 4.2.: Diagram of the methodology conducted in the model implementation and evaluation.

The model's evaluation may be found in sections 4.5 and 4.6. The confusion matrix of each model results was obtained to carry out the assessment. In this matrix the activities predicted by the model are compared with the actual activities expected.

Furthermore, several metrics such as global accuracy, and precision, recall, and F_1 for each of the activities have been computed. The model's overall accuracy is defined as the percentage of cases in which the model is correct. The ratio between the cases accurately predicted for an activity and the total cases predicted as that activity is called precision. While recall, on the other hand, assesses the proportion of properly forecasted instances against the number of cases that actually corresponded to that activity. Finally, the value of F_1 measures the balance between the two previous measures, precision and recall. The formula used to compute the value of F_1 is the following:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{4.1}$$

However, one cannot rely solely on these measurements of model performance to assess the correct functioning of the model. Because a black-box model could be obtaining these values correctly for the training and testing values, and yet not work properly and / or behave dangerously in other situations in which the model has not been tested.

In this research, we propose comparing the performance measures previously obtained against the explanations provided by the interpretability algorithms in order to trust the model and avoid incorrect evaluations.

LIME method will be used to contrast the performance measures. By approximating any classifier or regressor locally with an interpretable model, LIME is able to offer model-agnostic explanations to their predictions in a faithful way. It will be used in combinations with Submodular-Pick LIME, which is a method that uses submodular optimization in order to select a collection of representative examples to provide explanations. Therefore, the model behaviour can be explained in its whole domain.

4.3. Classical Machine Learning Approaches

4.3.1. Logistic Regression

Classical algorithms are the first strategy to be studied in this research. Despite the availability of increasingly complicated algorithms, conventional ML algorithms will continue to have a significant presence due to the parsimony principle, which asserts that the simplest solution that can explain the data should be chosen. Logistic regression is one of the simplest classification models. The aim is to use a sigmoidal curve to estimate the probability of a class. The sigmoidal function converts discrete or continuous data (x) into a numerical value (y) between 0 and 1.

This model will be fed with the features of the database described in the previous section. As indicated, only the mean, max and min values of those features relative to the time domain will be chosen (see Section 4.1 and 4.2 for more detail). And these data will be standardized before being used as input, using the StandardScaler function from the library Scikit-learn.

Using the logistic regression function of Scikit-learn, the logistic regression model, containing a single hyperparameter - the penalty - can be implemented. This hyperparameter can therefore imposes a penalty, if the model has too many variables. As a result, the coefficients of the less important variables would drop. In order to choose the accurate value of this parameter, several logistic regression models have been trained on the input data and cross-validated with different values of the inverse of the regularization strength (C) values including 1 or no penalty, 0.1, 0.01, 0.001, 0.0001, and 0.

The results of this validation, which can be seen in figure 4.3, indicate that the best value of C obtained for this specific dataset equals to 1, which is identical to not using a penalty. The model has been trained without penalty based on these results.

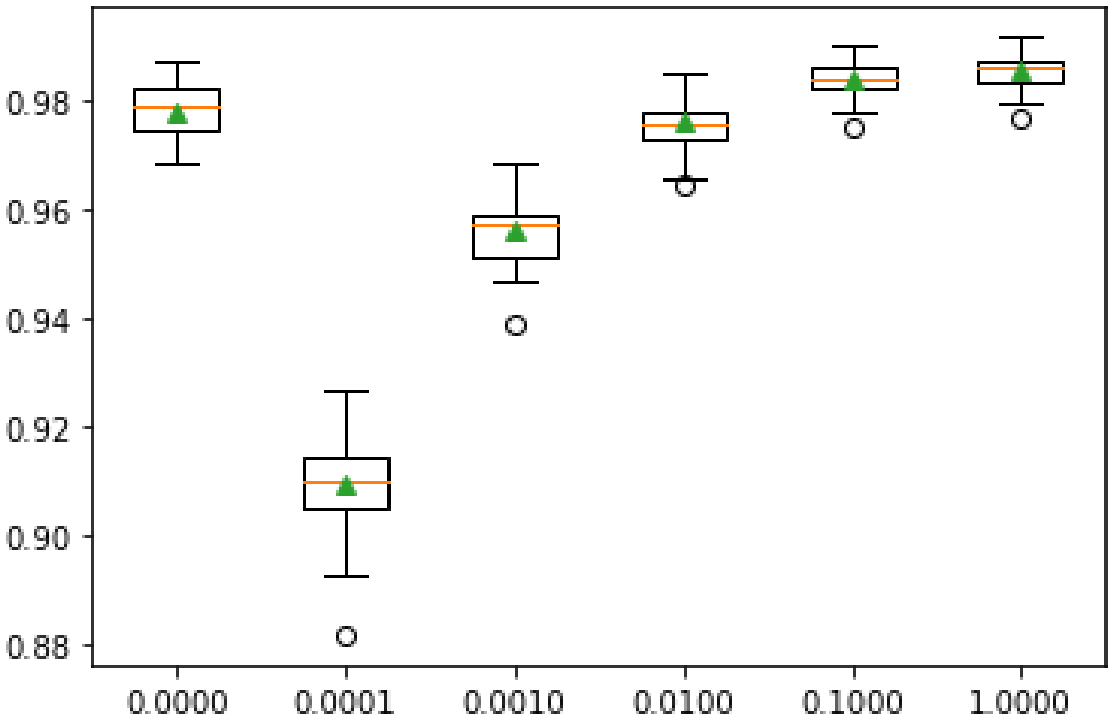


Figure 4.3.: Box plot comparing the scores (y-axis) of logistic regression models trained with different inverse of regularization strength values (x-axis).

The model's evaluation may be found in section 4.5 and 4.6. The confusion matrix of the model findings was obtained in order to conduct the evaluation. In this matrix, the model's predicted activities are compared to the actual activities expected. Furthermore, several measures such as global accuracy, precision, recall, and F_1 have been computed for each of the activities.

These model performance data, on the other hand, cannot be utilized to establish if the model is operating correctly. To ensure that the model is trustworthy and that erroneous judgements are avoided, we recommend comparing performance measurements to the explanations provided by interpretability algorithms like LIME and SP-LIME.

4.3.2. Decision Tree

More classical machine learning algorithms have been implemented in this study to assess the performance of simple models in human activity recognition applications. In the second approach a decision tree model will be explored. Because it is also an intrinsically interpretable algorithm, the decision tree has been picked as one of the standard machine learning algorithms to examine. Due to the simplicity of this model, decision trees are understandable and explainable models without the need for external tools. Decision trees are white-box models, so one can examine its functioning and how

the model makes achieves its results, in contrast with the information provided by the LIME algorithm.

This model will be fed by specific attributes from the Human Activity and Postural Transitions Dataset as specified in detail in sections 4.1 and 4.2.

The decision tree model is defined by two hyperparameters: maximum depth and minimum samples leaf. The number of nodes from the root note to the bottom of the model is indicated by maximum depth. The highest depth a decision tree can attain theoretically is one less than the number of training samples, but this should be avoided, as overfitting can occur. The minimal number of samples required to be at a leaf node is specified by the value of mininimum samples leaf. A split point will be considered if it leaves at least minimum leaf size of training samples in each of the left and right branches, regardless of depth. The model may be smoothed as a result of this.

The GridSearchCV function was used in order to choose the optimal values for max depth and min samples leaf, i.e., those hyperparameter values that optimize the model's accuracy. GridSearchCV is a scikit-learn class that allows you to evaluate and select model hyperparameters in a systematic manner. By indicating a model and the hyperparameters to test, it is possible to evaluate the first's performance in terms of the second's using cross-validation. It is a time expensive strategy, especially if a large number of variations hyperparameter variations want to be tested. However, GridSearchCV ensures that the model adjusts properly to the nature of the data. As the objective of this thesis is to examine both interpretability and accuracy, the hyperparameter search was limited to a meaningful range. This led to a significant reduction of the optimisation time.

In this approach, the hyperparameters values of max depth were permutated between 1 and 20 and min samples leaf between 1 and 20. Then the model's accuracy values were compared. Figure 4.4 shows the accuracy values achieved for these hyperparameter values. As shown in Figure 4.4, the value of the parameter max depth = 8, corresponding with the grey line, has the highest cross-validation scores. The difference between the values of the min_samples_leaf variable is not so easy to see with the naked eye, but the maximum score value would correspond to min samples leaf = 3. Therefore, the optimal decision tree model for our data has a maximum depth of 8 nodes, and the minimum number of samples to split an internal node is 3.

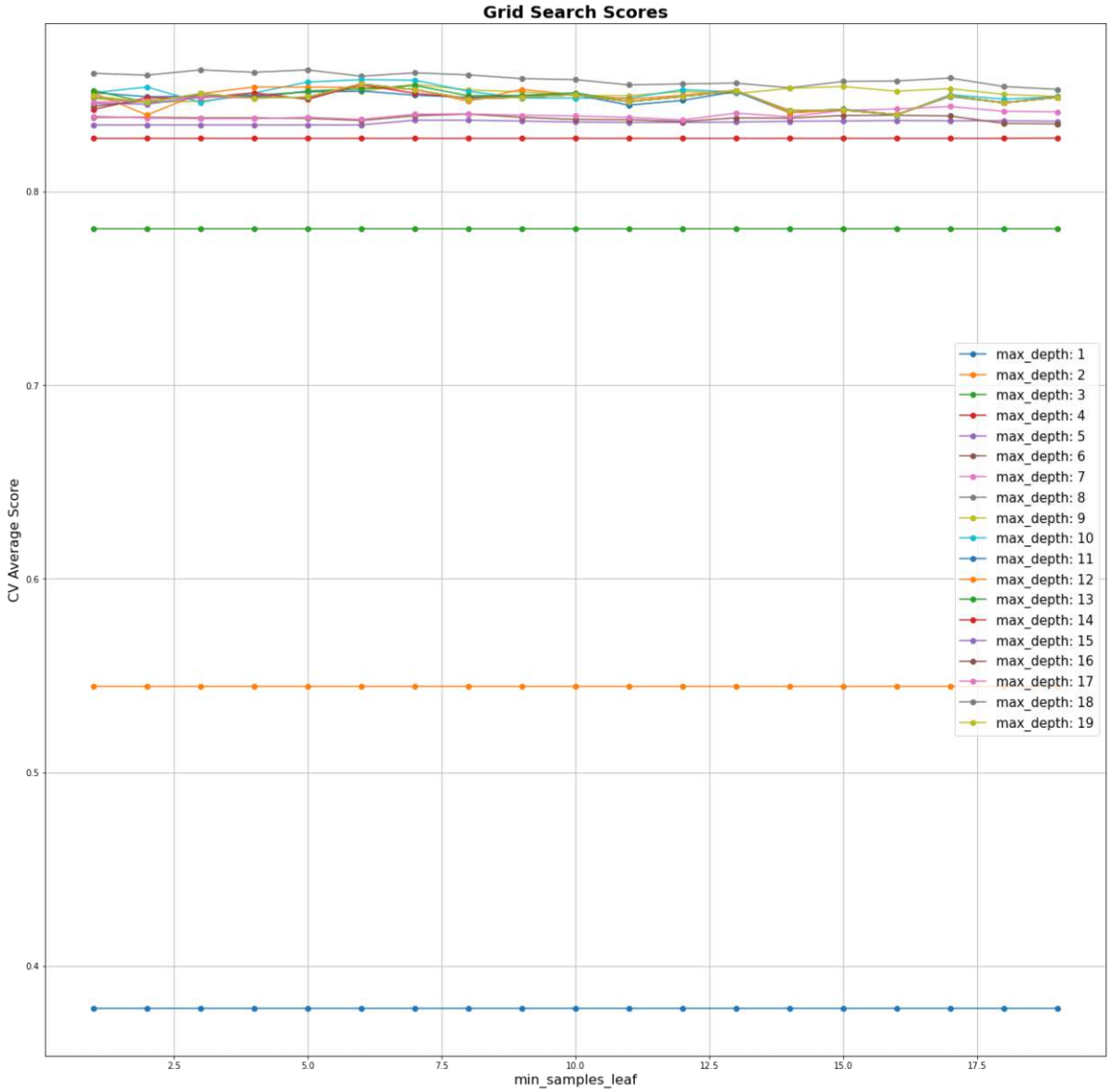


Figure 4.4.: Graph comparing the scores of Decision trees trained with permutations of the hyperparameters: max_depth and min_samples_leaf.

A schematic representation schematic representation of the optimal decision tree architecture of the model is presented in figure 4.5. In this tree the class of the samples after a decision split is indicated by colours. We can see how the first division made by the model separates the cases in which the activity being carried out is “laying” from the rest. Therefore, this activity will be easily identifiable by the model.

Moreover, the nodes on the left of the diagram belong to those related to the static activities “sitting” and “standing” which are not always differentiated adequately, but they are easily distinguished from the dynamic activities "walking" in orange colour and “walking upstairs”, and “walking downstairs” green both.

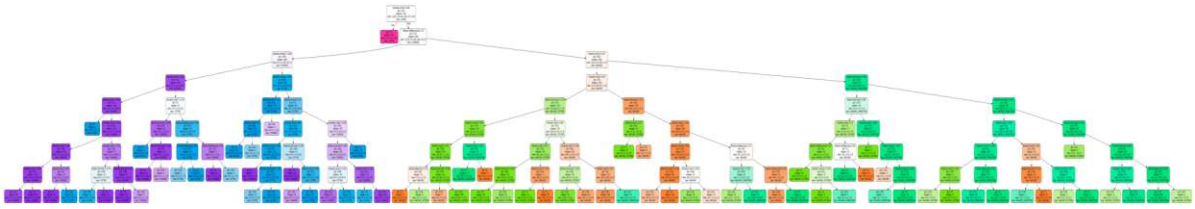


Figure 4.5.: Schematic of the Decision Tree Model architecture for Human Activity Recognition with max depth equals to 8 and minimum samples leaf equals to 3. The note corresponding to laying activity is presented in pink colour. Sitting and standing nodes are coloured blue and purple respectively. Orange nodes correspond to Walking instances, and green nodes to Walking upstairs and walking downstairs.

The evaluation of the model may be found in section 4.5 and 4.6. To conduct the assessment, the confusion matrix of the model findings was obtained. The model's forecasted activities are compared to the actual activities expected in this matrix. In addition, for each of the activities, numerous metrics like global accuracy, precision, recall and F_1 have been calculated. These performance metrics are then compared with the explanations obtained with LIME.

4.3.3. K-Nearest Neighbour

In this section, the last traditional machine learning algorithm of this thesis will be described. In this approach, the k-nearest neighbour (KNN) algorithm is used. This method was chosen because of its simplicity, which makes it a popular algorithm for human activity classification in the state of the art [42]. KNN, despite of being a straightforward technique, may frequently provide excellent results.

This model will be fed by the database features specified in section 4.1, as in the prior approaches. That is, only the mean, maximum, and minimum values of the time-domain features from the database will be selected, as previously stated for the two other traditional machine learning approaches (see section 4.1 for detail). Using the Scikit-learn library's StandardScaler function, this data will be normalized before being fed as input to the model. Furthermore, the y values corresponding to the activity performed, must be transformed to a single hot encoding before being fed to the model too.

The k-nearest neighbour classifier was implemented using scikit-learn. There are three parameters that determine this algorithm: p, n_neighbors, and leaf_size. P, is a value indicating which distance is going to be calculated. P=1 means using Manhattan distance, and p=2 corresponds to the Euclidean distance. N_neighbors indicate the number of neighbours required for each sample. And, leaf_size is given to the algorithm used to calculate the nearest neighbours. This can have an impact on the speed with which the tree is built and queried, as well as the amount of memory required to hold the tree. The best value is determined by the nature of the data.

In order to find the best values for the hyperparameters (n_neighbors, p and leaf_size), the GridSearchCV function was used, as in approach one. This function runs tests with the indicated hyperparameter permutations and finally it returns those hyperparameters that result in a model with better cross-validation score. It's a time-consuming method, especially if one wants to test a big number of hyperparameter modifications. GridSearchCV, on the other hand, ensures that the model adapts to the data effectively. As a model that that functions correctly well would suffice for the evaluation of interpretability, the algorithm has been evaluated with limited number of hyperparameter variations, reducing the time spent searching for the best model design.

The hyperparamater values' permutation was comprised between 15 and 30 for n_neighbour, and between 2 and 30 for leaf_size. The value of p has been set to 1. That is, Manhattan distance has been used, because better results have been obtained with this technique.

The model's evaluation may be found in section 4.5 and 4.6. The confusion matrix of the model results was obtained in order to conduct the evaluation. In this matrix, the model's predicted activities are compared to the actual activities expected. Furthermore, several measures such as global accuracy, precision, recall, and F_1 have been computed for each of the activities.

These model performance data, on the other hand, cannot be utilized to establish if the model is operating correctly. To ensure that the model is trustworthy and that erroneous judgements are avoided, LIME was used for the interpretability.

4.4. Deep Learning Approach

For the neural network approach that has been proposed in this thesis, recurrent neural networks based on Long Short-Term Memory cells have been used. The choice of LSTMs as a classification approach is highly suitable due to the temporal sequences of genuine human hand motions, as suggested by various papers presented in the state of the art. RNNs have the benefit of being able to make decisions based on current and previous inputs. The backpropagation update algorithm in LSTMs prevents the vanishing gradient problem in the training phase. Internal paths created by LSTM assist to retain errors for longer periods of time.

Convolutional and recurrent layers are combined in this design. The convolutional layers serve as feature extractors, providing abstract representations of sensor data in feature maps. The recurrent layers simulate the temporal dynamics of feature map activation.

Similar to the GridSearchCV function used in the previous three approaches, scikit has another function called RandomizedSearchCV. Both algorithms run tests with the indicated hyperparameter permutations and finally they return those hyperparameters that result in a model with better accuracy. In contrast to GridSearchCV, a defined

number of hyperparameter settings are sampled from the given distributions rather than all hyperparameter values being tried out.

This is a time-consuming procedure, especially if a large number of hyperparameter combinations is to be tested. Nevertheless, RandomizedSearchCV guarantees that the model is successfully adapted to the nature of the data. A this thesis examines the trade off between performance and accuracy, a model that works correctly will suffice. Therefore, the algorithm has been supplied with a limited number of parameter variants, decreasing the time spent looking for the optimum model design.

The implementation of the hyperparameter-tuning has resulted in a model architecture that combines convolutional layers with recursive layers and dense layers. The resulting architecture can be seen in Figure 4.6. The system includes four one dimensional convolutional layers with 256 filters and kernel size equal to 11. To reduce adjustment time, the hyperparameter values have been set the same for layers of the same type. The model contains two LSTMs layers that are fed the features extracted by the convolutional layers. Both LSTM layers have 200 units. The model includes a dropout layer with rate of 0.3. The dropout layer helps to minimize overfitting. This layer changes input units to 0 at random with a rate frequency at each step during training time. Inputs that aren't set to 0 are scaled up by 1/(1 - rate) so that the total sum remains the same. The dense layer has 100 units. The output layer, which is also dense, has 6 units, according to number of activities to be recognized. The activity probabilities for input data are obtained using a SoftMax function in the output layer. The identified human activity is the ultimate outcome.

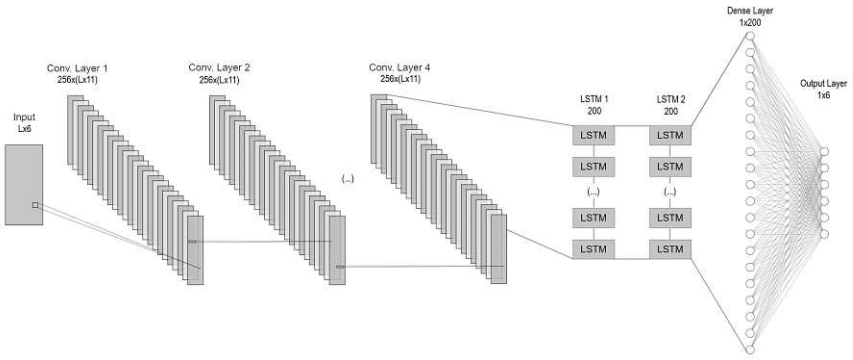


Figure 4.6.: Schematic from the proposed CNN-LSTM-based architecture for a human activity recognition system.

The neural network is fed a matrix of stacked time series data from the database of Reyes-Ortiz et al. [88]. In contrast to earlier techniques, the network will be given a raw data sequence. Only the body accelerometer and gyroscope data, not overall acceleration measurements, will be picked, as discussed earlier in section 4.1. The StandardScaler

function from the Scikit-learn library will be used to normalize input data before it is utilized as input. Prior to model training, the output y-values will be one hot encoded.

The evaluation of the model may be found in section 4.6. To complete the assessment, the confusion matrix of the model findings was obtained. The model's projected activities are compared to the actual activities expected in this matrix.

In addition, for each of the activities, numerous metrics like global accuracy, precision, recall, and F_1 have been calculated. The global accuracy of the model and the F_1 value are two values to which we will pay special attention because global accuracy informs us what percentage of situations the model accurately predicts. The F_1 number, on the other hand, tells us about the importance of precision and recall while also contrasting the balance of the two parameters.

However, these model performance data alone cannot be used to determine the model's right operation. Because the neural network as a black-box model may be obtaining the right values at random for the training and testing values, but then not work or act dangerously in other cases where the model hasn't been evaluated.

To trust the model and prevent inaccurate assessments, comparing previously acquired performance metrics to the explanations offered by the interpretability algorithms will be performed in this study. The performance measurements will be compared using the LIME approach. LIME is able to provide model-agnostic explanations to the predictions of any classifier or regressor in a faithful manner by approximating it locally with an interpretable model. It will be used in conjunction with Submodular-pick LIME, a method that use submodular optimization to choose a set of representative cases from which to deliver explanations. As a result, the model's behaviour can be explained over its whole domain.

4.5. Performance Evaluation

In this section, we will look at four HAR machine learning approaches that involve trust and understanding of predictions and models. Firstly, we assess the algorithms in particular by computing accuracy, precision, recall, and F_1 values, as well as displaying the confusion matrix, as is done traditionally. In the next section, LIME and SP-LIME will be utilized to provide explanations.

Other researches have proposed different methodologies for HAR using the UCI HAPT Dataset. For example, Zheng et al. [23] have proposed a model that reached an F_1-score for the global accuracy of 0.978. Zhang et al. [102] suggested the M-U-Net algorithm, which obtained accuracy results an F_1-score of 0.921 on the same database. And, LabelForest, which was proposed by Ma et al. [103], achieved accuracy results of a global F_1-score of 0.932 using also the same database.

The **logistic regression** algorithm trained on 60 time-domain features achieves a global accuracy of 89%. The activity recognized with the best F_1 value is "laying". While the remaining activities show a similar ease to be recognized. In the confusion matrix of the figure 4.7, it is possible to observe that there is a greater ease between the activities "Walking", "Walking upstairs", and "Walking downstairs" to be confused with each other. And there is also a tendency for "sitting" and "standing" to be confused with each other. Nevertheless, there are 21 instances standing out, because the behaviour of "laying" has been mistaken with the activity of "walking upstairs". The table contains the computed precision, recall, and F_1 values.

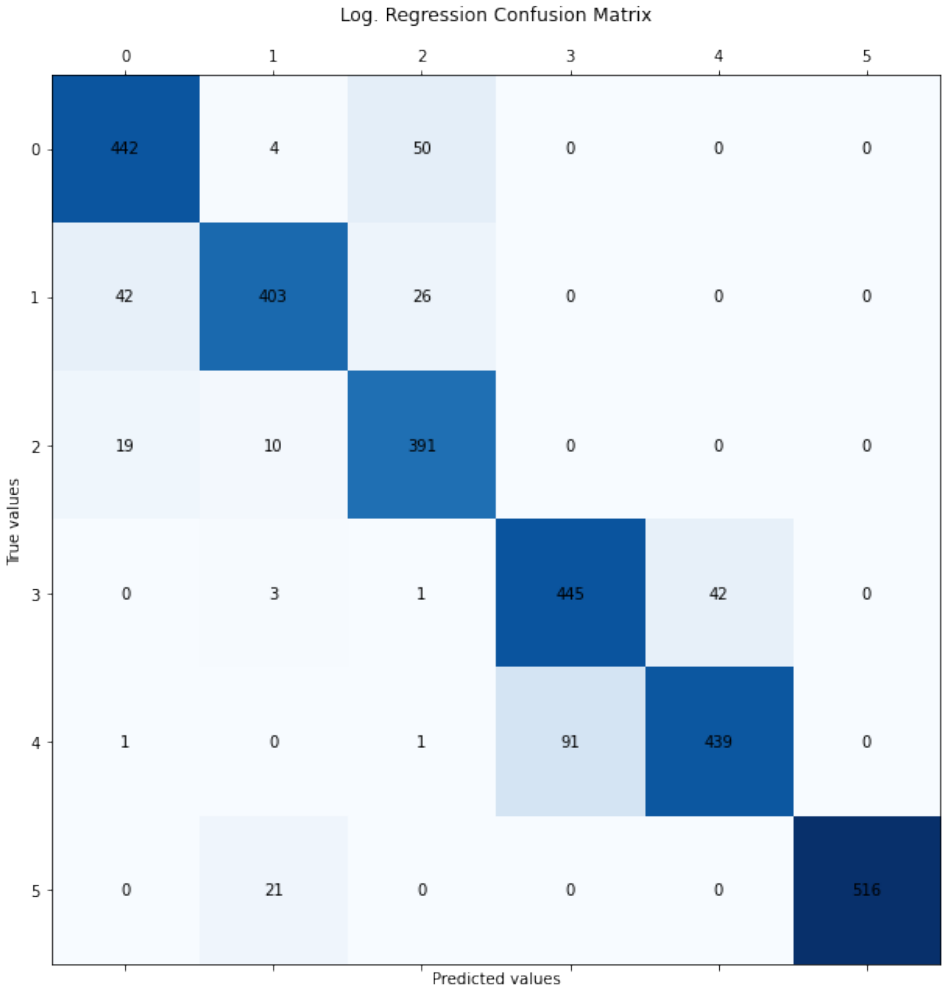


Figure 4.7.: Confusion matrix showing the results of the Logistic Regression's activity recognition.

	precision	recall	F_1-score	support
WALKING	0.88	0.89	0.88	496
WALKING UPSTAIRS	0.91	0.86	0.88	471
WALKING DOWNSTAIRS	0.83	0.93	0.88	420
SITTING	0.83	0.91	0.87	491
STANDING	0.91	0.83	0.87	532
LAYING	1.00	0.96	0.98	537
Global accuracy			0.89	2947

Table 4.1.: Performance scores of approach one - logistic regression.

The second approach has been evaluated using a similar methodology. The **decision tree model** obtains an overall accuracy of 80% after being trained using the 60 features of the temporal domain.

The error patterns between the activities are identical to the ones from the preceding approach, as shown in the confusion matrix (see Fig. 4.8). The recognition of instances corresponding to walking, walking upstairs and walking downstairs activities are often mistaken with each other. The same thing happens as well with some static activities such as sitting and standing. However, in this method, instances corresponding to laying are 100% successfully recognised.

Because the decision tree is not a black box model, and its structure was presented in Figure 4.5, more insight can be obtained from this model prior to LIME explanation's analysis. As seen in the model diagram, instances corresponding to the "laying" action were the first to be detected by the model, and the examples related to it were instantly segregated from the others according to the value of the "tGravityAcc-min()-X" feature.

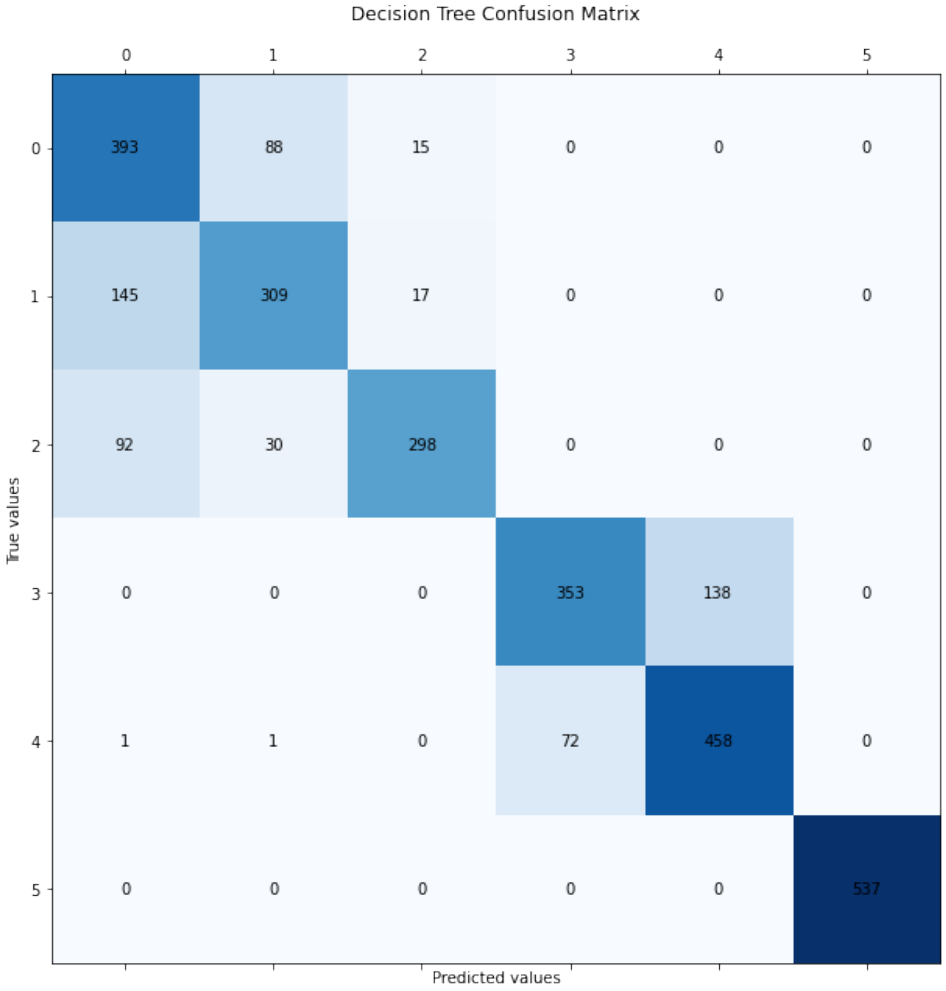


Figure 4.8.: Confusion matrix showing the results of the Decision Tree's activity recognition

	precision	recall	F_1-score	support
WALKING	0.62	0.79	0.70	496
WALKING UPSTAIRS	0.72	0.66	0.69	471
WALKING DOWNSTAIRS	0.90	0.71	0.79	420
SITTING	0.83	0.72	0.77	491
STANDING	0.77	0.86	0.81	532
LAYING	1.00	1.00	1.00	537
Global accuracy			0.80	2947

Table 4.2.: Performance scores of approach two - decision tree.

After training the **K-Nearest Neighbour algorithm** model with the 60 time-domain characteristics, which corresponds to the third strategy, this model achieves an accuracy of 85%.

In addition, it can be seen in the confucion matrix (see Fig 4.9) how, in comparison to the prior methodologies, the action “walking” is less wrongly predicted as “walking upwards” or “walking downstairs” in the confusion matrix. Also, “walking downstairs” is never mistakenly predicted as “walking upstairs”. Static behaviours such as “sitting” and “standing” remain frequently mistaken as one another. Finally, “lying” is a 100% successfully predicted activity.

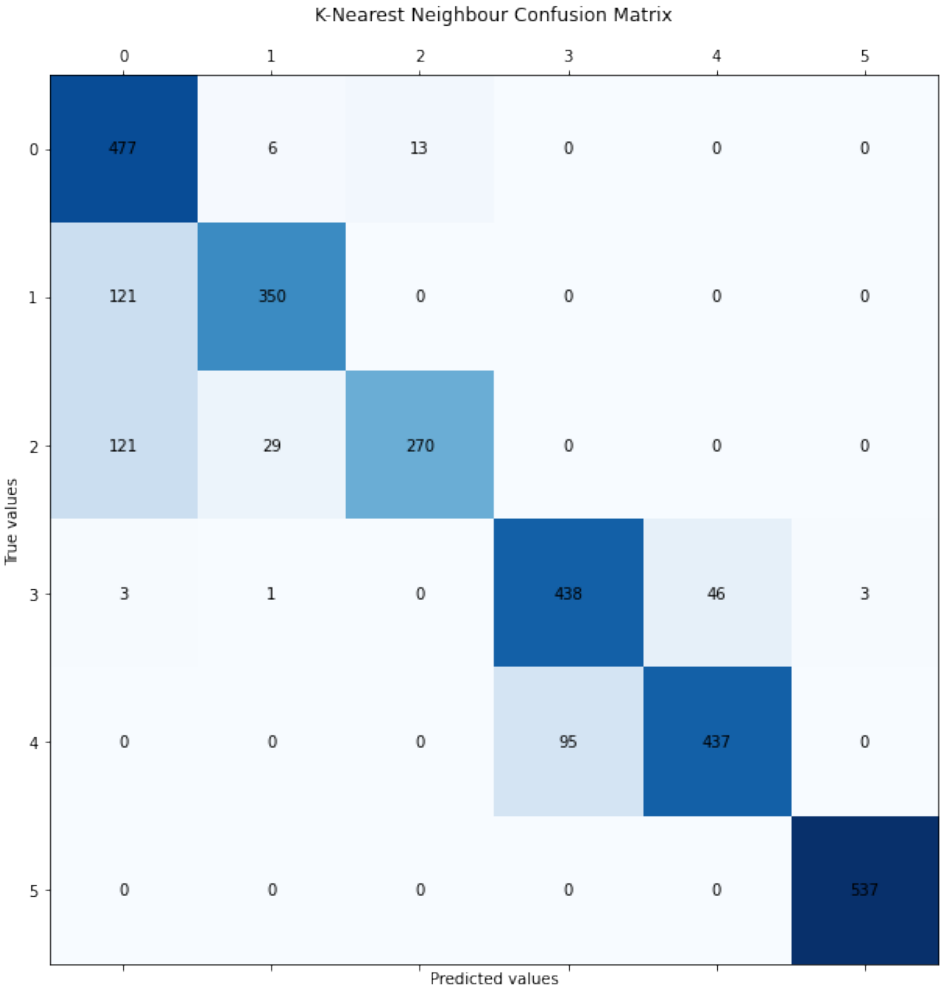


Figure 4.9.: Confusion matrix showing the results of the K-Nearest Neighbour’s activity recognition.

	precision	recall	F_1-score	support
WALKING	0.66	0.96	0.78	496
WALKING UPSTAIRS	0.91	0.74	0.82	471
WALKING DOWNSTAIRS	0.95	0.64	0.77	420
SITTING	0.82	0.89	0.86	491
STANDING	0.90	0.82	0.86	532
LAYING	0.99	1.00	1.00	537
Global accuracy			0.85	2947

Table 4.3.: Performance scores of approach three - k-nearest neighbour.

The **convolutional and recurrent neural network** is the last approach that was studied in this thesis. This proposed architecture, which was trained with raw sensor data obtained from the database, obtains a global accuracy of 92%. In comparison to the confusion matrices produced from the preceding models, the confusion matrix of this model (see fig. 4.10) shows that there are a significantly fewer number of mistakes between the recognition of "walking", "walking upstairs", and "walking downstairs" activities.

The identification between "sitting" and "standing' instances appears to have improved in comparison to prior models, however its recognition error still exists. What stands out from this confusion matrix is the mistake between the static activities ("sitting", "standing", and "laying"). While the previous recognition models were able to identify "laying" instances with almost exact accuracy and F_1 values greater or equal to 0.98. This neural network obtains a similar accuracy to that of the other static activities, among which it is sometimes confused when predicting.

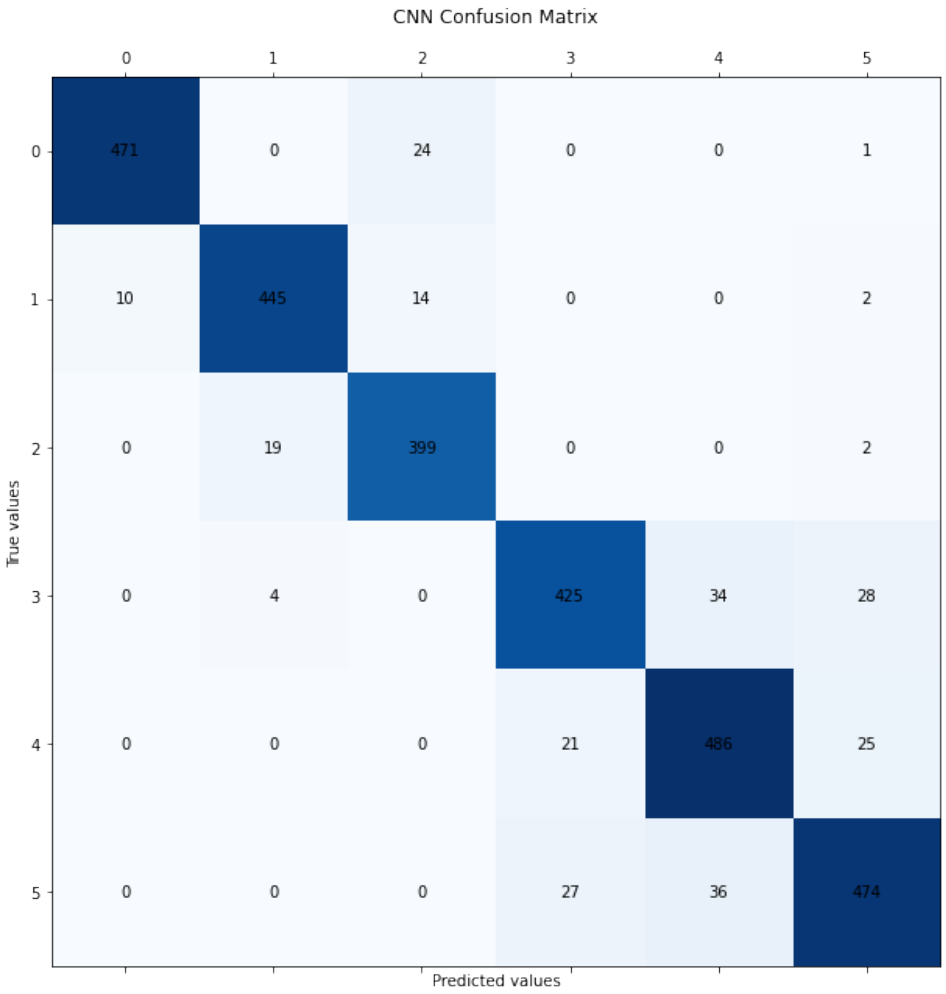


Figure 4.10.: Confusion matrix showing the results of the Convolutional and Long Short-Termed Memory's activity recognition.

	precision	recall	F_1-score	support
WALKING	0.98	0.95	0.96	496
WALKING UPSTAIRS	0.95	0.94	0.95	471
WALKING DOWNSTAIRS	0.91	0.95	0.93	420
SITTING	0.90	0.87	0.88	491
STANDING	0.87	0.91	0.89	532
LAYING	0.89	0.88	0.89	537
Global accuracy			0.92	2947

Table 4.4.: Performance scores of approach four - convolutional and long short-termed memory neural network.

The neural network presented in this study outperforms traditional machine learning algorithms in global accuracy, as seen in the comparison graph of F_1-scores from each

activity recognition performance table (see Fig. 4.11). This convolutional- and LSTM-based algorithm obtains higher F_1 value for the recognition all the studied activities except for "laying", in which traditional models show better performance. For example, the decision tree and the K-nearest neighbour model both obtain an F_1 value of 1 for the recognition of "laying" instances, while the decision tree gets the poorest performance values for the recognition of the rest of the activities.

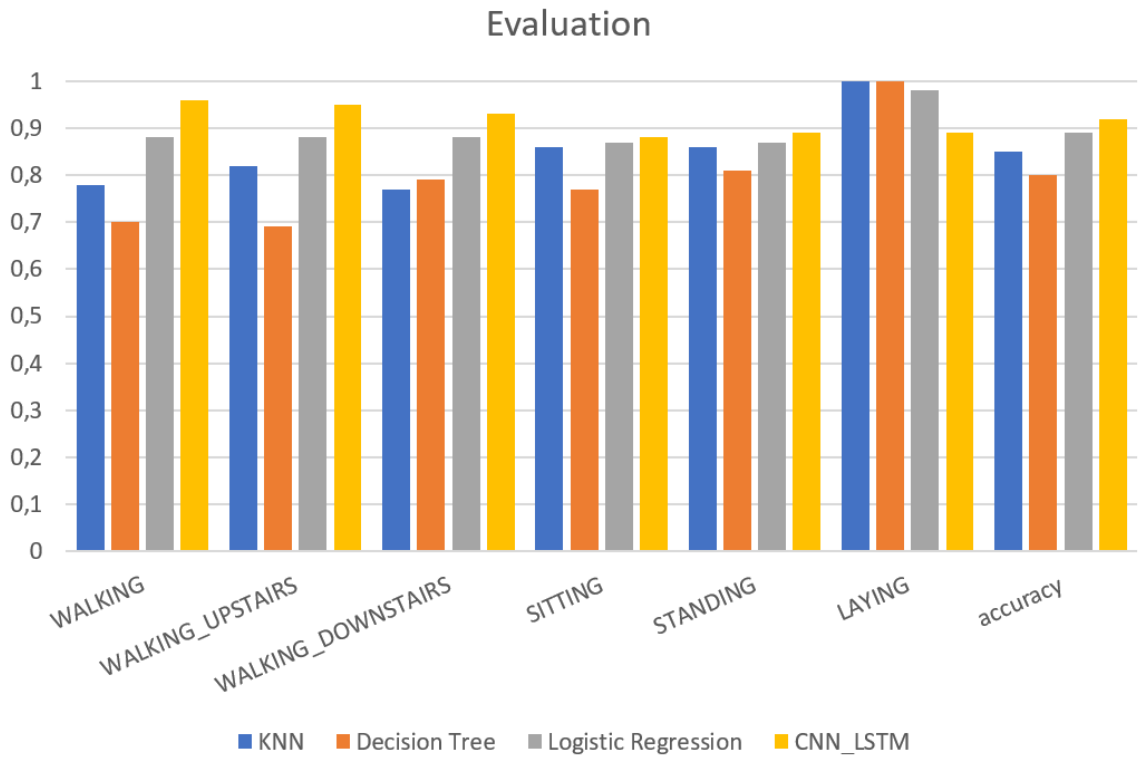


Figure 4.11.: Graph showing performance F_1-scores (y-axis) for recognition of each human activity (walking, walking upstairs, walking downstairs, sitting, standing, and laying) and global F_1-score for each approach.

4.6. Interpretability Evaluation

Logistic Regression

Due to legibility reasons, only three outputs for the logistic regression model for HAR using SP-LIME and LIME (Figure 4.12) are presented in this section as an example, while twelve further explanations are shown in Appendix A. On the x-axis, each feature contribution to the prediction probability is shown by a bar. Each feature's bar is coloured green for positive values and red for negative values. The names of significant

features are displayed on the left. Those features that are more relevant for recognizing the action in the title are displayed in a sorted order depending on their importance.

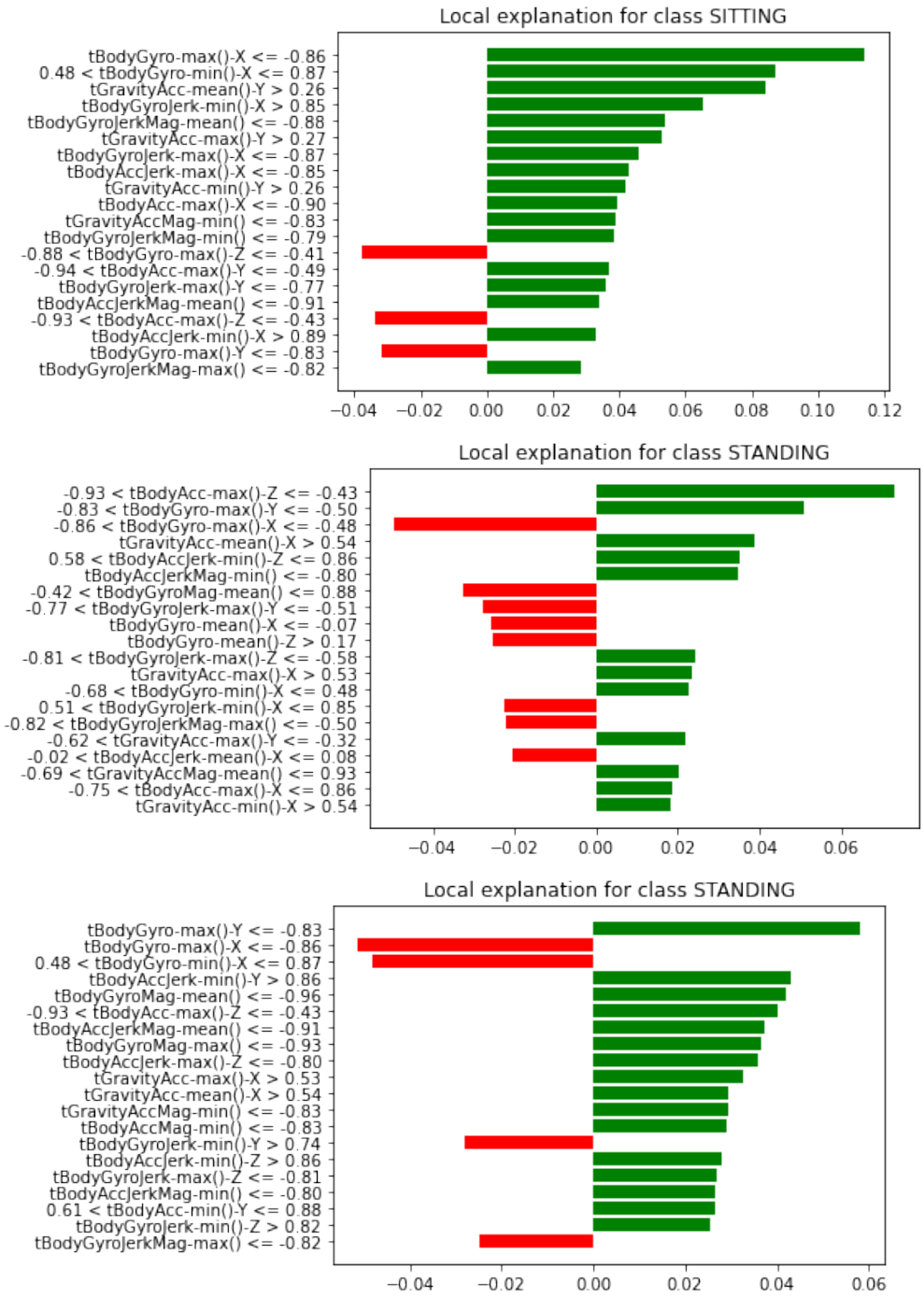


Figure 4.12.: Local explanations for three significant instances provided by LIME and Submodular-Pick LIME for the logistic regression algorithm.

In general, the algorithm's values of each feature's contribution to the probability of

correctly predicting the corresponding activity are quite low. Perhaps the algorithm's activity recognition mistakes are due to the low contribution values assigned to each feature. This might be the reason why the instances belonging to the "sitting" activity were mistaken with the corresponding instances of the standing activity and vice versa as was seen when we examined the confusion matrix (see Fig. 4.7). The high values of features with negative contribution to probability (shown in red) and the absence of greater positive probability values than those which contribute negatively, as seen in Fig. 4.12 (a) for example, is also remarkable and it can potentially contribute to the mistake discussed earlier.

Furthermore, the SP-LIME algorithm mostly considers sitting and standing cases as relevant examples for interpretation, with only one explanation provided from a non-standing or sitting example.

From the explanations from figure 4.12, it can be seen that the features that contribute the most to the recognition of the activity "standing" are *tBodyAcc-max()-Z* $\in$ *(-0.93, -0.43]*, *tBodyGyroMag-mean()* $\in$ *(-0.96, -0.42]*, *tBodyGyro-max()-Y* $\in$ *(-0.83, -0.5]*. However, these above-mentioned features, only contribute to the probability with significance values ranging between 0.05 and 0.07. We propose to compare these features with the ones that contribute the most to the recognition of "sitting" instances, which are *tBodyGyro-min()-X* $\in$ *(0.48, 0.87]*, *tBodyGyro-mean()-X* $\leq$ *-0.07*, *tBodyGyroJerk-min()-X* $\in$ *(0.51, 0.85]*, *tBodyGyro-max()-X* $\in$ *(-0.86, -0.48]*.

The explanations provided for sitting and standing classes share some significant features. The feature's contribution to one class prediction or the other is determined by the feature's range of values. Therefore, it is possible that if the feature range is not accurate when its values change, the probability of the instance will be incorrectly attributed to another class, as it happens in figures 4.12 (a), (b), and (c) with the feature *tBodyGyro-max()-X* $\in$ *(-0.86, -0.48]*.

The three features that contribute the most to walking instances' recognition are: *tBodyAcc-min()-X* $>$ *0.54*, *tBodyAccJerkMag-mean()* $\in$ *(-0.77, 0.89]*, and *tGravityAcc-mean()-X* $>$ *0.54*. However, as previously stated, the presence of so many features with negative contribution to its recognition (shown in red) among the ones with highest contribution values in the explanation causes us to be sceptical of this model. It is risky to assign values that often appear as features with highest positive contribution to the recognition of other classes. This might explain the recognition mistakes and numerous failures when analysing the confusion matrix.

Decision Tree Model

Due to legibility reasons, only three outputs for the logistic regression model for HAR using SP-LIME and LIME (Figure 4.13) are presented in this section as an example, while twelve further explanations are shown in Appendix B. In some of the explanations shown in figures 4.13 (a) and (c), the prediction probabilities are coloured green for positive values and red for negative values on the x-axis. On the left, the names of

prominent features are displayed. The elements that are more important for recognizing the action in the title are shown in descending order of priority.

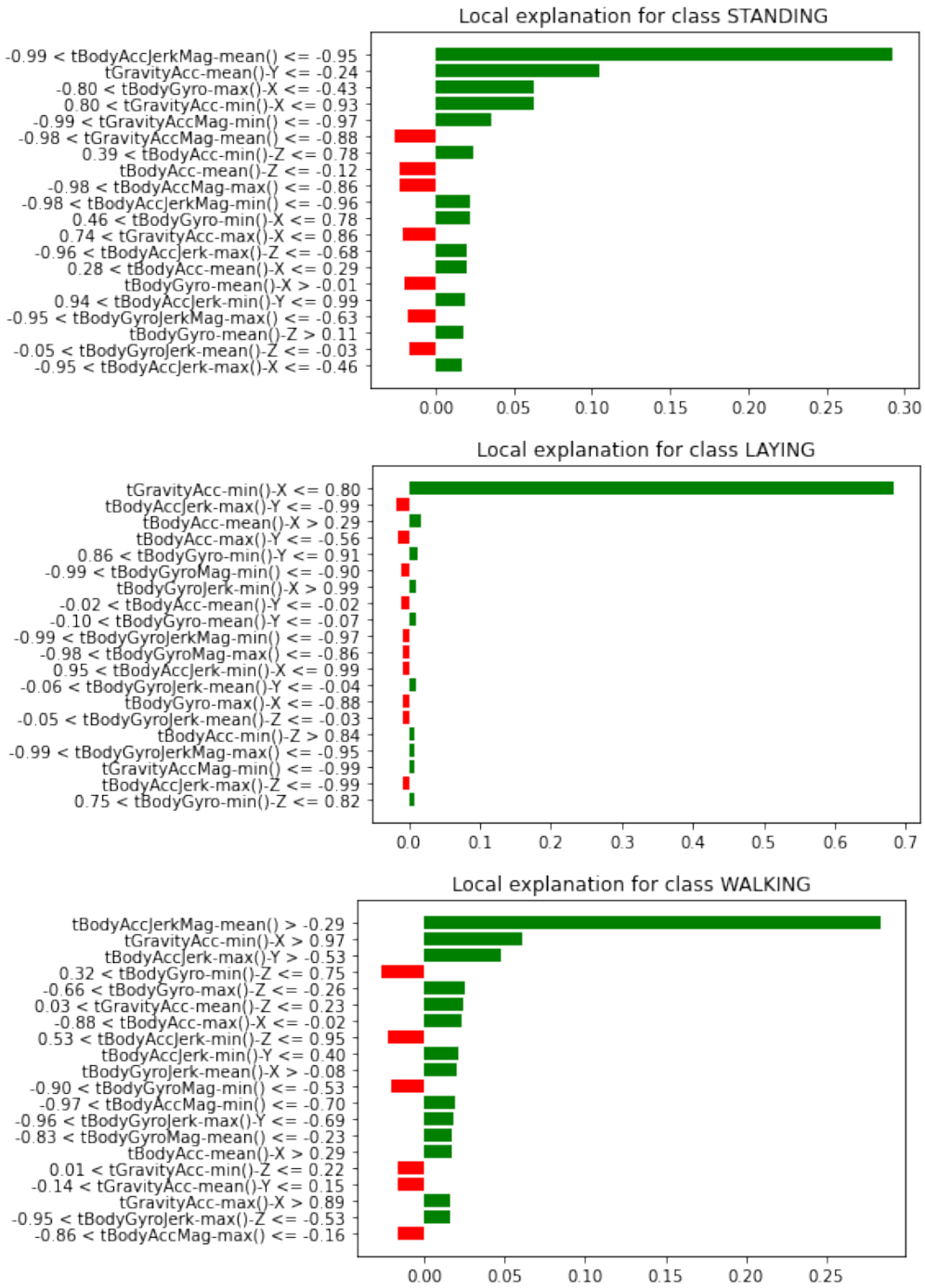


Figure 4.13.: Local explanations for three significant instances provided by LIME and Submodular-Pick LIME for the decision tree algorithm.

The explanations of this algorithm have features with very high probability values

compared to the explanations obtained from the other traditional algorithms. Moreover, as could be seen on the decision tree's architecture shown in Fig. 4.4. The features used to make the decisions are similar, shown as branch splits on the schematic.

The decision tree scheme from figure 4.4 indicated that the activity being carried out was classified as laying when the values received from the feature *tGravityAcc-min()-X* $\leq$ *0.096*. As expected, this feature appears as significant for "laying" instances recognition in figure 4.13 (b). However, LIME's explanations results show us that the instances that have a value of *tGravityAcc-min()-X* $\leq$ *0.8* have more than a 60% probability of being the "laying" activity. The condition provided by the algorithm is less restrictive than the one known from the model architecture. However, by looking at the model's confusion matrix (see Fig. 4.8) and seeing that F_1 equals 1 for this activity, it can be determined that this does not cause any problem in terms of recognition.

In the architecture of the decision tree model (see Fig. 4.8), the feature *tBodyAccJerkMag-mean()* splits between dynamic and the remaining static activities. If its value is lower or equals to -0.79 the model classifies those instances as "sitting" or "standing". And if the value is greater than -0.79 it corresponds to walking activities.

As expected, this feature also appears in the LIME algorithm's descriptions of the remaining actions. In the explanation obtained with LIME, the division is executed at the feature value of -0.95 instead of in -0.79. However, looking at the model's confusion matrix, we can observe that just two instances of static activities are labelled as dynamic. This implies that the model's performance is mostly unaffected by this change.

However, it should be noted that after this split, there are a high number of errors between activities of the same category. This might be caused because the feaures with greater contribution to "walking" recognition's probability are *tBodyAccJerkMag-mean()* $>$ *-0.29* and *tBodyAccJerkMag-mean()* $\in$ *(-0.95, -0.29]*. The first one matches the feature with greater contribution to "walking downstairs" recognition, and the second one to "walking upstairs recognition". Also, instances corresponding to the classes "standing" and "sitting" also share the feature *tBodyAccJerkMag-mean()* $\in$ *(-0.99, -0.95)*.

Despite the fact that this method provides poorer performance results than the approach 1 model (see Fig. 4.11), LIME's explanations for this model are more easily understandable. The findings of the prediction are based on fewer criteria, but they have a greater degree of significance. As a result, the explanations are more informative, and a model like this may even give the user greater confidence.

Last curious aspect about this model is that the factors that have the most impact on the recognition probability are those gathered only from the accelerometers of the smartphone. This might suggest that if the data supplied by the phone's gyroscope was ignored, just a little amount of information relevant to recognition would be lost. We suggest to obtain a compromise between data-privacy and accuracy, since if less data

was be used, e.g., using only accelerometer data in this case, then user's data privacy would be enhanced.

K-Nearest Neighbour

Due to legibility reasons, only three outputs for the logistic regression model for HAR using SP-LIME and LIME (Figure 4.14) are presented in this section as an example, while twelve further explanations are shown in Appendix C. On the x-axis, prediction probabilities are coloured as green for positive values and red for negative values. The names of notable features are listed on the left. In descending order of priority, the components that are more significant for recognizing the action in the title are indicated.

These explanation's probability values are often low. This might be due to the confusing nature of the probability concept in a K-Nearest Neighbour model. However, the probability values of the explanations provided by the algorithm for cases matching to the class lying, for example, have a maximum value of 0.035. While looking at the confusion matrix and performance numbers of this method, we can see that it has an F_1 score of 1 and a 100% accuracy.

Moreover, excluding the case of Walking downstairs instances, the explanations typically highlight a few features that are obviously more important than the others. As a result, when evaluating the K-Nearest Neighbour model, we will focus on these most significant factors that contribute the most to the prediction.

The features that contribute the most to "walking" instances are: *tBodyAccJerk-max()-Z* > 0.63, *tBodyGyroJerkMag-min()* > 0.65, *tGravityAccMag-mean()* $\in$ *(-0.69, 0.93]*, *tBodyAccMag-mean()* $\in$ *(-0.69, 0.93]*. Some of these features, especially the last two, are closely related to the most significant features of other activities. To put it another way, the feature *tGravityAccMag-mean()* has been considered important for instance recognition as "walking" up to 0.93. However, when the max value of this same feature (*tGravityAccMag-max()*) is greater than 0.83, the instance can be considered as "walking downstairs".

And exactly the same thing happens with the second most significant feature of the explanations of the instance corresponding with the class "walking downstairs", which is *tBodyAccMag-max()* > 0.83. It is possible that this problem may be the cause of the mistake when recognizing the downstairs classes that are sometimes recognized simply as walking. Thus, it is also possible that these factors are also present in instances belonging to the walk upstairs class, even though the most relevant features for this class recognition, according to the explanation that the algorithm has provided, are *tBodyGyroMag-min()* > 0.66 and *BodyGyroMag-mean()* > 0.88.

The cases of sitting and standing also share the most important features with which they are recognised. But these features have distinct values in each case. For sitting instances' recognition the most significant features are: *tGravityAcc-max()-Y* > 0.27, *tGravityAcc-min()-Y* > 0.26, and *tGravityAcc-mean()-Y* > 0.26. And, for the standing

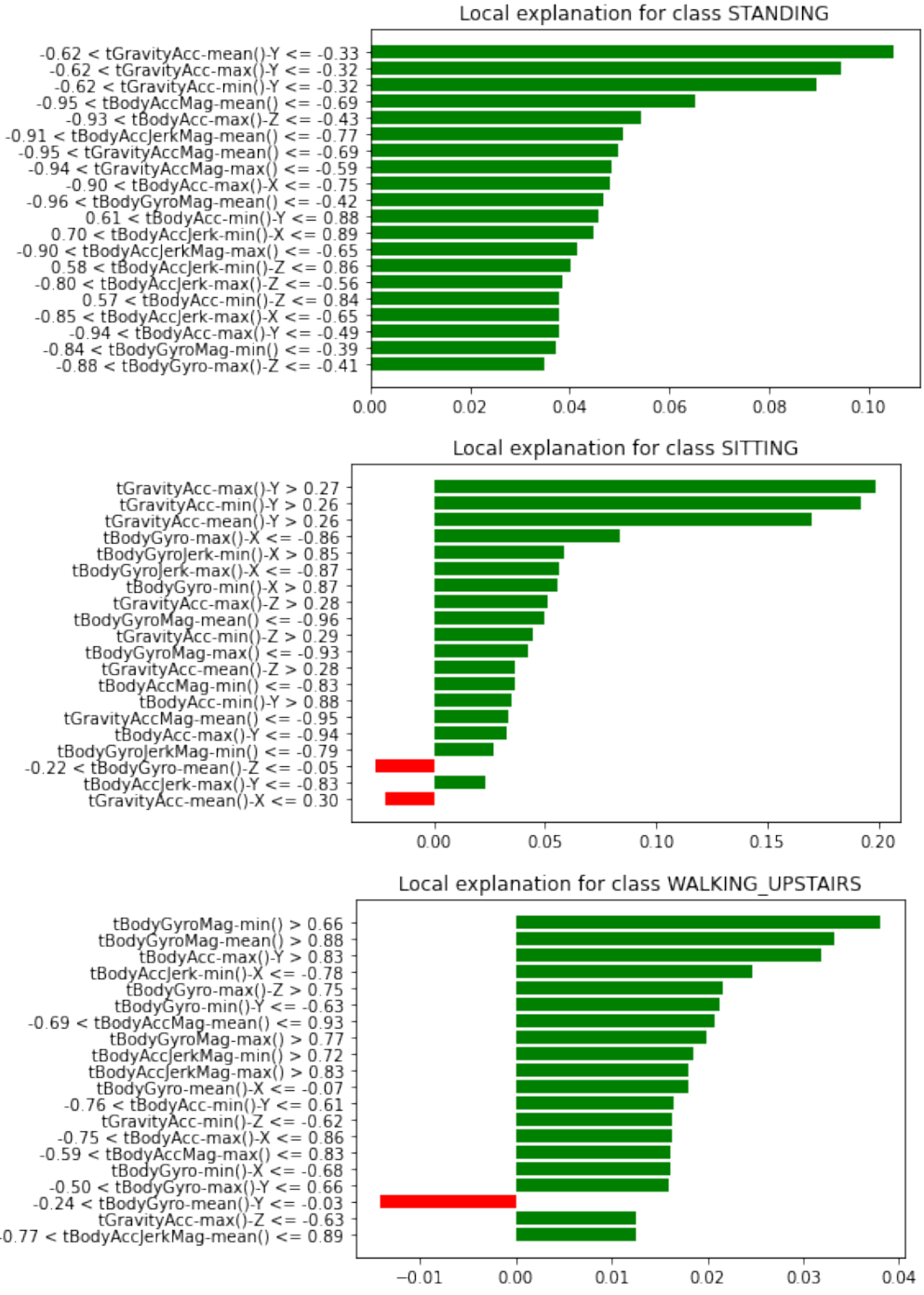


Figure 4.14.: Local explanations for three significant instances provided by LIME and Submodular-Pick LIME for the k-nearest neighbour algorithm.

instances the features that contribute the most to its recognition are: *tGravityAcc-min()-Y ∈ (-0.62, -0.32]*, *tGravityAcc-max()-Y ∈ (-0.62, -0.32]*, and *tGravityAcc-mean()-Y ∈ (-0.62, -0.33]*.

The model's behaviour would be more trustworthy, and mistakes would be avoided, if the recognition was determined by totally different features. For example, the features with the most significance in "laying" activities are:. tGravityAcc-mean()-X $\leq$ 0.3, tGravityAcc-min()-X $\leq$ 0.28, and tGravityAcc-max()-X $\leq$ 0.3.

CNN-LSTM Neural Network

Due to legibility reasons, only three outputs for the logistic regression model for HAR using SP-LIME and LIME (Figure 4.15) are presented in this section as an example, while twelve further explanations are shown in Appendix D. Prediction probabilities are colour-coded on the x-axis as green for positive values and red for negative ones. On the left, the names of noteworthy features are mentioned. The features that are more important for recognizing the action in the title are listed in descending order of priority.

Because this model's input data is a three-dimensional time series of raw data, the explanations generated by LIME differ from the ones from the previous approaches. Therefore, the explanations cannot be analysed in the same way as in the preceding situations. The LIME algorithm assesses the precise values of each input feature differently depending on the corresponding time interval. In order to make a distinction of the time instant, a number was added to the end of each feature.

The time at which the input feature occurs intuitively should not be important for activity recognition. Since, in each situation or depending of the person, the specific movements of the sensorics and the values registered by the sensor could not occur at the same time for the same activities. Nevertheless, these raw data has been subsequently filtered inside the model by some convolutional layers in order to extract more significant features with which the following neural network's layers have been fed. As a result, the data offered by LIME and SP-LIME's algorithm explanations will be analysed, and it will be determined although the value of the moment of time might not provide much information for our evaluation.

Furthermore, because of this time-instant distinction, there is a lot of diversity across the model features, each of them has a lower relevance and contribution to the likelihood of each class. As a result, the explanations provided by LIME and SP-LIME's algorithms should be assessed in a different way in this approach.

We can see that the explanations related to dynamic activities like "walking" and "walking downstairs", which have the greatest F_1 and accuracy scores in this model, do not have any features in red that contribute negatively to its recognition among the twenty most important features shown in the explanations. While explanations for static activities such as "standing" or "lying" contain features, and even sometimes they are the most significant features of said explanation, with negative values that do not contribute to the recognition of this activity. This might make us distrust the model's capacity to distinguish actions like these, which, if we look closely, correlate with those activities that had the poorest F_1 scores in table 4.4.

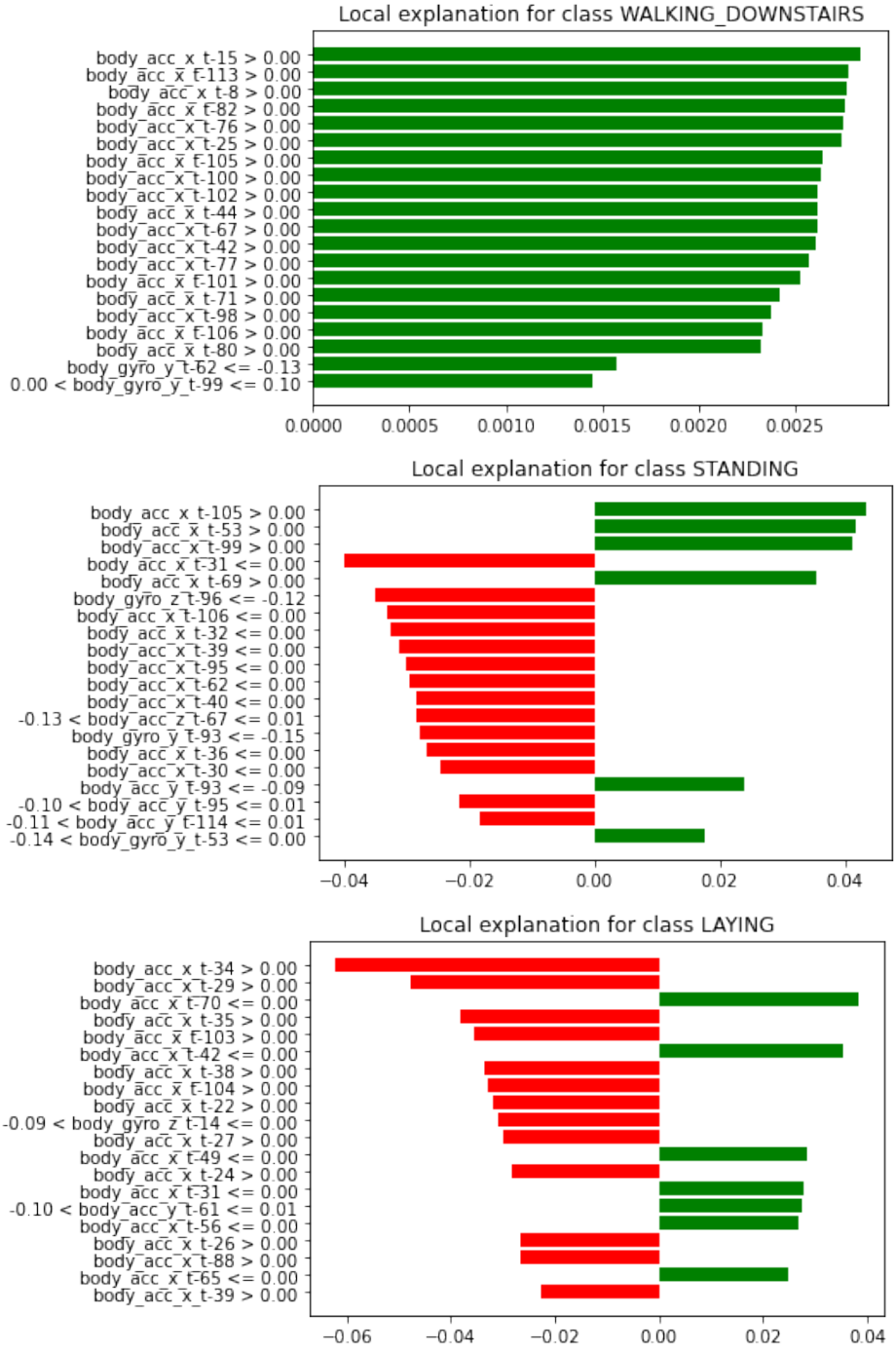


Figure 4.15.: Local explanations for three significant instances provided by LIME and Submodular-Pick LIME for the convolutional and long short-termed memory neural network algorithm.

Furthermore, it very remarkable that the great majority of the features featured in the twelve explanations of this model are collected by the accelerometer, specifically the

values corresponding to the x-axis of this sensor. If this algorithm is indeed capable of properly predicting the proposed human activities using only the accelerometer's x-axis readings, it would be a significant step forward in terms of the privacy of the data acquired by these sensors in these applications.

4.7. Results and Further Opitimisation

Using the results from the preceding practical section, the goal of this section is to propose improvements in the design of human activity recognition models for human-centred cyber-physical systems. The enhancements to be described in this section are intended to achieve a compromise between interpretability and accuracy.

A first proposal that has previously been employed in this study is selecting and using only databases with easily interpretable features for training and testing machine learning algorithms. For example, in section 4.2, it was mentioned that only features that belonged to the temporal domain will be used. This was done because they are easier to understand than those from the frequency domain. Furthermore, just the most basic statistical values, i.e., minimum, maximum, and mean, were selected so that the generated explanations were also more comprehensible by the worker or the production manager.

In this initial proposal, we can see the trade-off between interpretability and accuracy that was mentioned above. As a result of the feature selection, the models of the classical ML approaches that were supposed to be fed with 561 features from the UCI HAPT database were only fed 60 features. In most cases, having fewer information results in getting a less accurate model. However, transparent and interpretable models are also required. Understanding the model's behaviour and increasing user confidence need interpretability. This is especially important for safety-critical applications.

For example, the explanations generated by the interpretability algorithm in the previous section can lead to dubiousness in case of logistical regression, because there were so many features with such a huge negative contribution to the recognition. When these explanations are compared to those offered by the LIME algorithm for the decision tree model, the decision tree's explanations are considerably easier to understand and sometimes only rely on one or two significant features. In comparison to the logistic regression model, these explanations offered us a lot more confidence and reliability in the model. This can be said even despite the fact that the model's performance scores were lower.

Furthermore, some legal issues, such as GDPR's "right to an explanation," have emerged as a result of the recent awareness in privacy and cybersecurity. Regarding this concern, Explainable Artificial Intelligence can assist with this task. In light of the above, enhancing users' data privacy, for example, is another proposed way to improve the design of models for human activity recognition.

XAI helped to determine the inputs that contribute the most to the recognition of each action for each model thanks to the explanations provided in section 4.6. There, we observed that the features that contributed the most to the recognition for the decision tree and neural network were those gathered from the data collect by the smartphone's accelerometer. Not only that, the most significant features in the Deep Learning approach are those corresponding only to the sensor's X-axis data.

This is especially important, as privacy is coupled with determining the identity. According to Rasnayaka and Sim's study [71], IMU authentication algorithms can be used to recognise identity as well as several attributes that can impact worker's vulnerability. Thus, leaving out important input features or data, e.g. y-axis values, can inhibit the performance of identification methods.

As a result, it is suggested to achieve a balance between the privacy of the user's data and the precision of actions recognition by utilizing less information from the sensors. To accomplish this, the performance scores from both models will be compared using all of the data collected by the sensors on the one hand, and a smaller amount of data on the other.

Decision tree

Figure 4.16 illustrates the confusion matrix that results from using a decision tree model to recognize the same activities feed the model with the 27 features generated from accelerometer-collected data. We can see that the results obtained by this data-privacy proposal are very similar to the ones obtained in section 4.5, despite only using 45% of the information that was previously being used. The comparison of F_1-scores is shown in figure 4.17.

The improvement in user privacy stands out among the benefits that this decrease of information would provide. Therefore, it is strongly suggested to use only the features provided by the data collected by the accelerometer in order to reduce the collection of user information. Accepting lower performance in recognition of the activities in return for improved user privacy would even be justifiable.

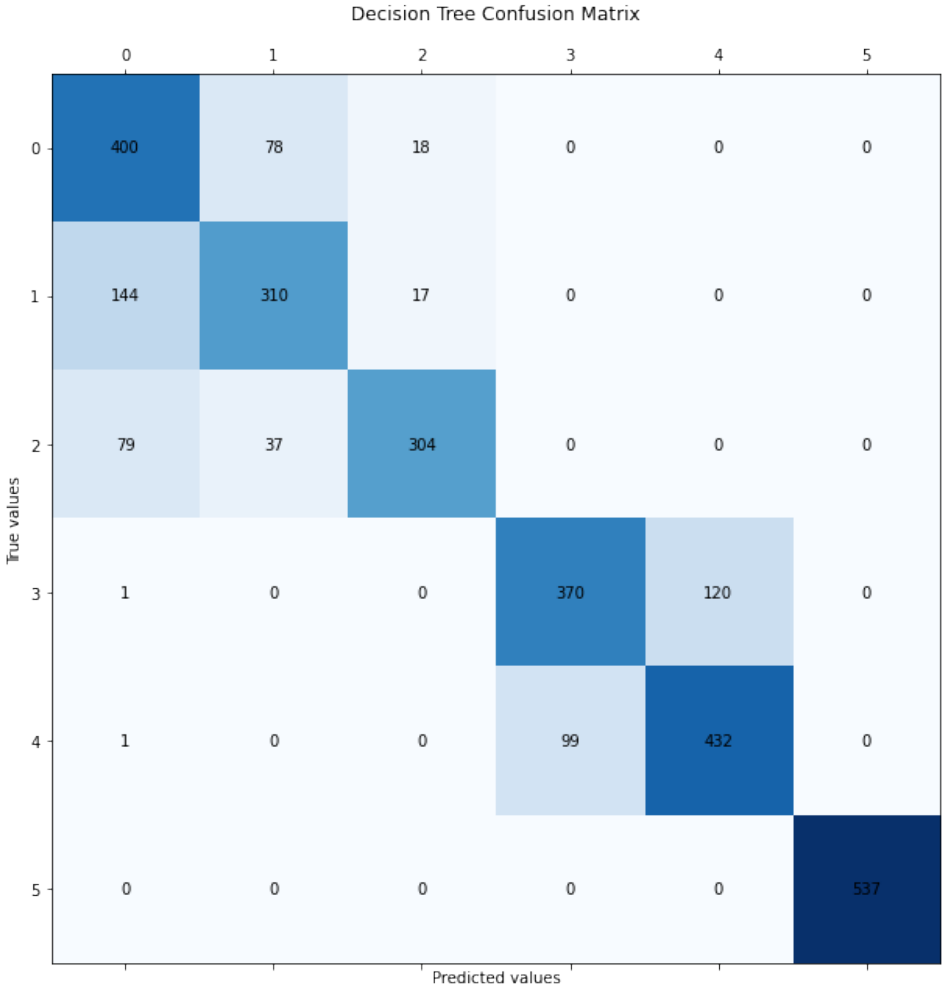


Figure 4.16.: Confusion matrix showing the results of the Decision Tree's activity recognition using only accelerometer-collected data.

	precision	recall	F_1-score	support
WALKING	0.64	0.81	0.71	496
WALKING UPSTAIRS	0.73	0.66	0.69	471
WALKING DOWNSTAIRS	0.90	0.72	0.80	420
SITTING	0.79	0.75	0.77	491
STANDING	0.78	0.81	0.80	532
LAYING	1.00	1.00	1.00	537
Global accuracy			0.80	2947

Table 4.5.: Performance scores of the Decision tree model using only accelerometer-collected data.

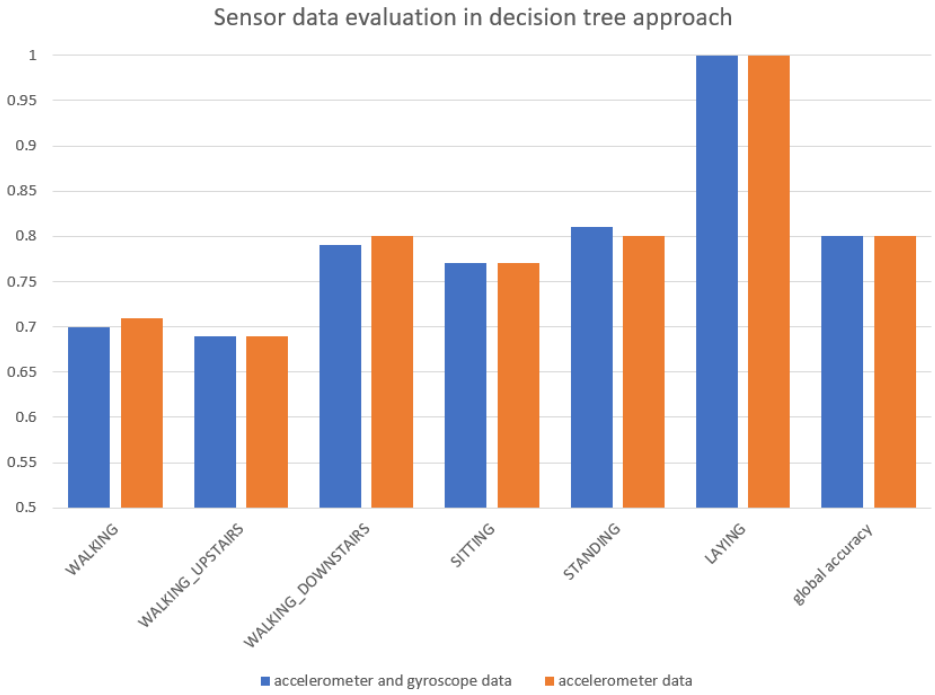


Figure 4.17.: Graph showing performance F_1-scores (y-axis) for recognition of each human activity (walking, walking upstairs, walking downstairs, sitting, standing, and laying) and global F_1-score for each decision tree approach.

CNN LSTM NN

The results of the model fed with all sensor data will be compared to the results of the model using only the accelerometer data and the model using only the X-axis data of the accelerometer sensor to examine the suggestion of information reduction in the deep learning approach.

Two distinct behaviours emerge from the findings of the suggested improvement for this method. On the one hand, the information reduction has performed badly for static activity recognition. The recognition of this activities (sitting, standing, laying) performed slightly worse even when the algorithm included every sensor. And the explanations given by LIME and SP-LIME (see figure 4.14) revealed numerous features that had a negative contribution to its recognition. It seems that many of the factors that have a large negative impact to its recognition have been kept, resulting in extremely low F_1 scores.

However, the recognition of dynamic activities such as “walking,” “walking upstairs,” and “walking downstairs,” works perfectly both with the data reduced to accelerometer data and with the data reduced to only the x-axis data. The latter means using only a 16.7% of the data initially proposed in the first approach.

As a result, this proposed improvement is not advised for the detection of static

activities and would be only recommended if the application requires dynamic activity recognition.

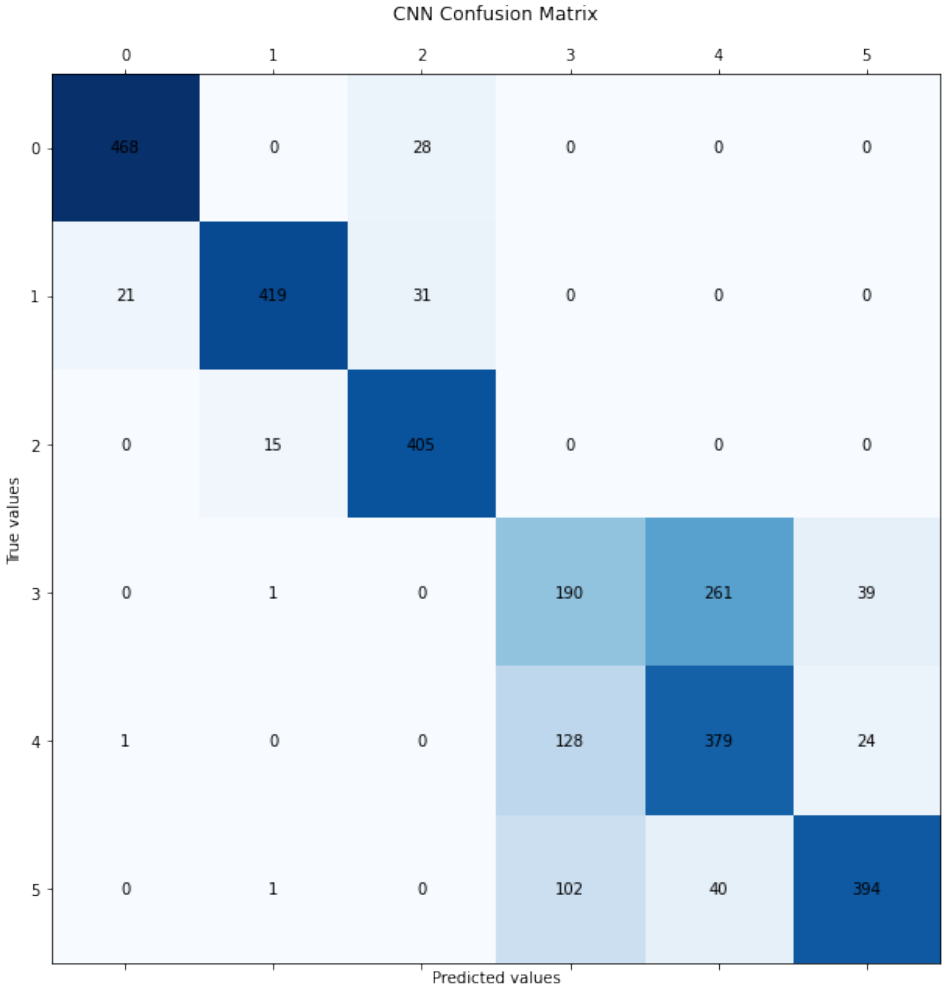


Figure 4.18.: Confusion matrix showing the results of the Deep Learning approach's activity recognition using only accelerometer-collected data.

	precision	recall	F_1-score	support
WALKING	0.96	0.94	0.95	496
WALKING UPSTAIRS	0.96	0.89	0.92	471
WALKING DOWNSTAIRS	0.87	0.96	0.92	420
SITTING	0.45	0.39	0.42	491
STANDING	0.56	0.71	0.63	532
LAYING	0.86	0.73	0.79	537
Global accuracy			0.77	2947

Table 4.6.: Performance scores of the Deep Learning model using only accelerometer-collected data.

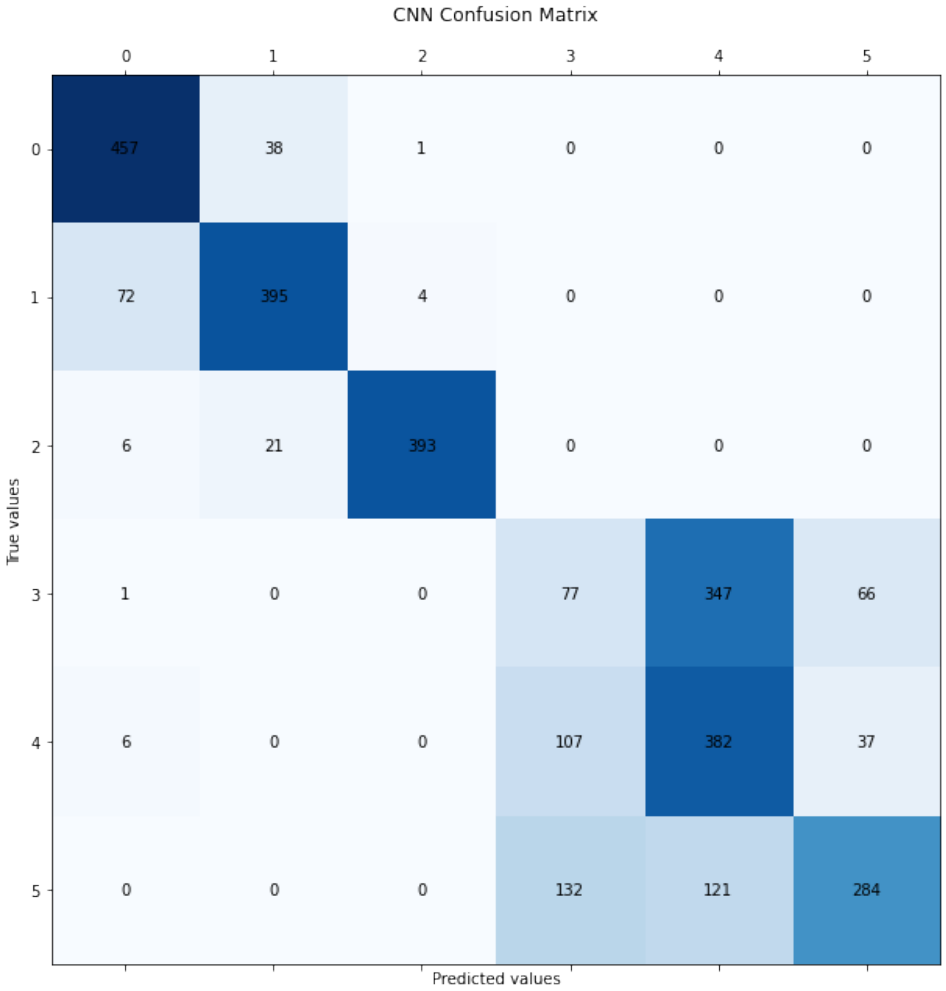


Figure 4.19.: Confusion matrix showing the results of the Deep Learning's activity recognition using only x-axis accelerometer-collected data.

	precision	recall	F_1-score	support
WALKING	0.84	0.92	0.88	496
WALKING UPSTAIRS	0.87	0.84	0.85	471
WALKING DOWNSTAIRS	0.99	0.94	0.96	420
SITTING	0.24	0.16	0.19	491
STANDING	0.45	0.72	0.55	532
LAYING	0.73	0.53	0.61	537
Global accuracy			0.67	2947

Table 4.7.: Performance scores of the Deep Learning model using only x-axis accelerometer-collected data.

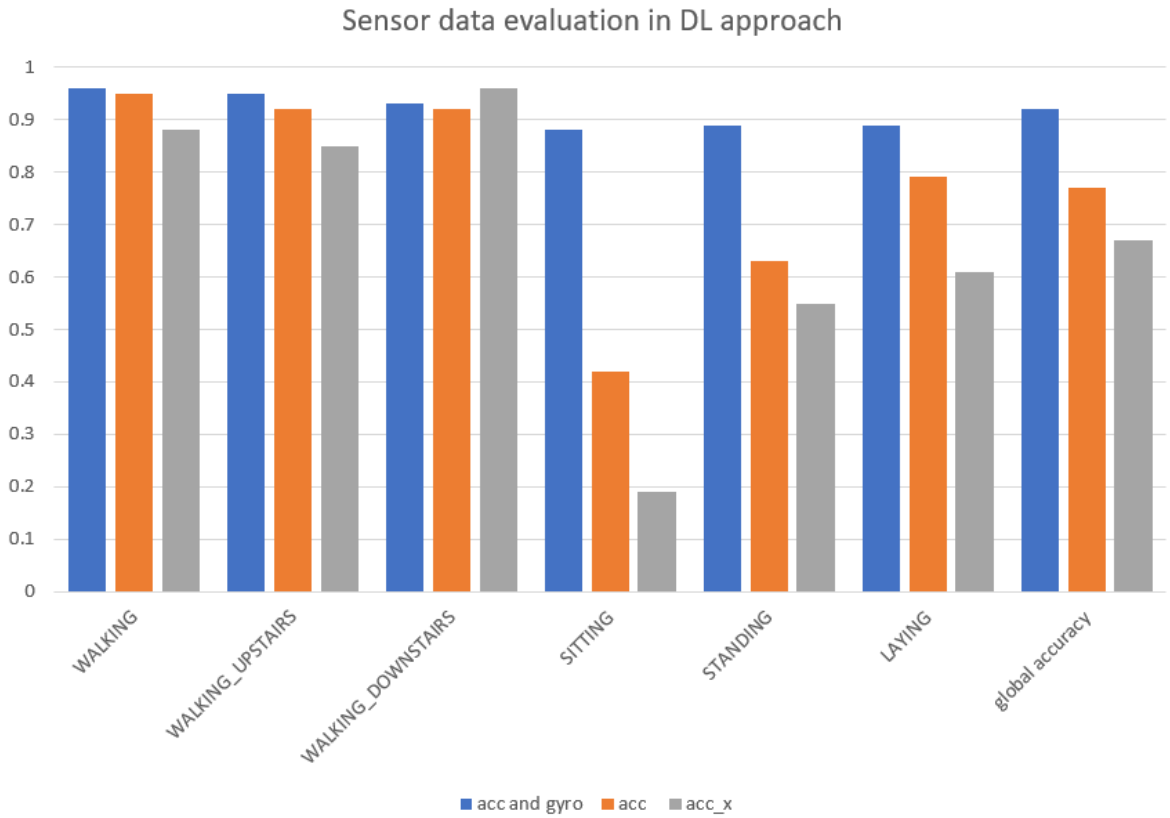


Figure 4.20.: Graph showing performance F_1-scores (y-axis) for recognition of each human activity (walking, walking upstairs, walking downstairs, sitting, standing, and laying) and global F_1-score for each deep learning approach.

5. Conclusion and Outlook

In this last chapter, we will delve into the thesis' results. A description of the findings and their significance is presented. Then, the research questions will be answered to verify or reject the hypothesis presented in the first chapter. Afterwards, the findings' relevance will be assessed while also placing the thesis in the perspective of other relevant research found in the literature review. The last section will focus on laying the groundwork for future studies in this and related topics.

This work presents a method for using explainable Artificial Intelligence in models used for Human Activity Recognition (HAR). As the Industry 4.0 is not migrating towards workerless production facilities, HAR is used to sense and analyse additional information about the worker to optimise processes and provide relevant assistance. Therefore, HAR is gaining importance in cyber-physical production systems (CPPS).

By using LIME and Submodular-Pick LIME explanation algorithms, the main difference between the proposed HAR methodology and the state-of-the-art HAR is that the interpretability of the model allows for further analysis when the system is failing or how to justify the decision-making process. Although machine learning algorithms appear to be accurate in forecasting, they are not without errors. The most crucial is the lack of transparency or accountability, which is inherent with black box ML models.

Some researchers consider interpretability an important indicator for the model, not only in terms of task performance but also in terms of auxiliary requirements such as security, privacy, non-discrimination, fairness, avoidance of technological debt, reliability, provision of the right to explanation, trust, etc. This is especially important, as privacy is coupled with determining the identity. According to Rasnayaka and Sim's study [71], IMU authentication algorithms can be used to recognise identity as well as several personal characteristics that can impact worker's vulnerability. Thus, within the methodology presented in this thesis, it is proposed to use the information provided by the XAI to efficiently reduce the amount of input data, in order to obtain a balance between privacy and accuracy in the HAR model.

Activity recognition has been performed for sensor-derived data in this work, as sensor-based monitoring is immune to extraneous disturbances that might confuse and distort the data obtained. This thesis, in particular, makes advantage of data generated by smartphones. As noted in the state-of-the-art study, there are a number of reasons why a smartphone is an excellent instrument for detecting human behaviours, e.g., its low-cost device able to combine several software and hardware sensors, cloud computing capabilities, etc.

Three different classical machine learning algorithms and a neural network combining convolutional and recurrent layers have been used to evaluate the methodology. The methodology presented in this paper achieves the desired balance between interpretability and accuracy for classical machine learning models. For these models, accuracy values of up to 89% are achieved, in addition to the explanations of the models' outputs. The developed neural network achieves higher accuracy values that are closer to those obtained by other researchers' HAR models.

5.1. Conclusion on the Research Questions

Furthermore, specific enhancements are recommended for both the neural network and the decision tree to reduce the amount of input data without significantly decreasing the performance, with the objective of achieving the aforementioned balance between privacy and accuracy.

We will continue by looking over the research questions and assessing the results that have been presented in this thesis:

1. **What is interpretability in machine learning and why is it important for human-centred production?**
   Extracting patterns, understanding causes for decisions, boosting confidence in model decisions, and discovering mistakes and overfitting are some of the benefits of XAI that have been examined. Interpretability of the models helps us to guarantee that legal and ethical considerations are satisfied, which is important given recent cyber-security and privacy concerns.

   However, aside from the legal constraints imposed by the GDPR, interpretability is critical because in many applications, just a prediction is insufficient. The desire for interpretability and explanations is motivated by factors such as making sense of the operation in order to validate and assess its security, and to improve social acceptability of the algorithm.

   The challenges and risks of using a black box model, especially in a human-operated setting who might be harmed by the algorithm's judgements, has increased the demand for interpretable machine learning models. Today's cyber-physical production systems are heavily monitored with sensors and may gather user data, moreover, decisions produced by these models might have an impact on worker integrity.

2. **How can the explainability and interpretability of models typically used for HAR be achieved?**
   The method for achieving model interpretability differs based on the approach

employed. From the review of the literature on interpretability and HAR carried out in this thesis, state-of-the-art techniques were evaluated and it was selected which would be employed later in this work.

Vision and/or sensor-based hardware are typically utilized in HAR applications. Particularly Inertial Measurement Units(IMUs), are commonly employed because to current pricing and their availability. IMUs can be found in a variety of consumer electronics and industrial applications, including smartphones, smartwatches, and other wearable devices.

Classification methods such as SVM, random forest classifier, k-nearest neighbour, and neural networks are typically utilized. In the literature, two approaches are distinguished depending on the algorithm employed. On the one hand, the literature distinguishes between transparent models, which are inherently interpretable due to their fundamental nature, such as decision tree models or linear models. More sophisticated models, on the other hand, can be explained using external XAI tools, which is known as the post-hoc explainability approach.

LIME (Local Interpretable Model-agnostic Explanation), Saliency Maps, and RISE are the most well-known model-agnostic post-hoc explainability tools. Using LIME any classifier's predictions can be explained. Saliency maps represent graphically the most representative inputs. RISE creates a significance map that depicts the relative importance of each pixel in the final forecast.

3. **How can XAI help to improve the design of HAR models for use in work assistance systems?**
   This last question about improving the design of HAR models for cyber-physical production system applications using XAI was the most important. Three different traditional machine learning models and one deep learning strategy were implemented and assessed for this question. The deep learning model is based on a successful combination of convolutional and recurrent layers that has previously delivered cutting-edge results in other time-series domains such as voice recognition.

   The explanations for these four approaches were obtained using the LIME and Submodular-Pick LIME explainable post-hoc tools. These algorithms allowed to identify potential areas for improvement for each model by examining the explanations provided.

   On the one hand, among the proposals for improvement, we suggest a selection of the model's input parameters that are easier to comprehend by the user. This ensures that the explanations obtained by LIME might be as easily interpretable as possible. Although we lose input data and therefore accuracy in the prediction,

the aim is to achieve a trade-off between the interpretability of the model and its performance. In light of the explanations gained, another solution has been proposed: a selection of fewer information to increase data privacy. Numerous vulnerable attributes for the worker can be obtained through IMU data. As a result, taking out crucial input features or data, such as gyroscope-collected data, might inhibit the performance of identification algorithms.

Using only 45% of the sensor-collected information for the decision tree model, the solution suggested in this thesis was able to achieve similar development results. Also, excellent performance values were obtained for dynamic activities detection, such as walking, walking upstairs, and walking downstairs, with the improvement proposed for the deep learning approach which employed only 16.7% of the data.

Both enhancements are intended to provide a repeatable way for improving human activity recognition tasks utilizing similar sensors. We intend to encourage its adoption in order to obtain a better trade-off between recognition performance, model interpretability, and worker data privacy.

The information shown above aids in a better comprehension of this thesis contribution. The findings should be considered while attempting to strike a balance between interpretability and accuracy in a machine learning model, particularly in Human Activity Recognition, as well as being able to enhance the model owing to the data privacy explanations offered.

5.2. Outlook and future work

Since this thesis was focused on investigating only human activity applications in which wearable sensors provided by a mobile phone with an accelerometer and gyroscope were used. Further research should definitely focus on employing this replicable methodology for diverse applications.

The technique described in this thesis is adaptable to a variety of settings. And, as a future research project, it is suggested to employ visual sensorics or a mix of visual and wearable sensors to perform human activity detection tasks.

Although, a model is proposed in this work, there are state-of-the-art models in the literature that have extremely excellent performance. The methodology proposed in this research might be applied to those more complex models, in order to improve the compromise between accuracy and interpretability.

List of Figures

List of Tables

Bibliography

[1] H. Lasi, P. Fettke, H. Kempe, T. Feld, and M. Hoffmann, "Industry 4.0," *Business and Infor-mation Systems Engineering*, 2014.

[2] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhartand, O. Sauer, hG.Schuh, W.Sihn, and K.Ueda, "Cyber-physical systems in manufacturing," *CIRP Annals*, 2016.

[3] E. Roth, M. Moncks, T. Bohne, and L. Pumplun, "Context-aware cyber-physical assistance sys-tems in industrial systems: A human activity recognition approach," *Proceedings of the 2020 IEEE International Conference on Human-Machine Systems, ICHMS 2020*, 2020.

[4] C. Jobanputra, J. Bavishi, and N. Doshi, "Human activity recognition: A survey," *Procedia Computer Science*, 2019.

[5] A. Bibal and B. Frénay, "Interpretability of machine learning models and representations: An introduction," *ESANN 2016 - 24th European Symposium on Artificial Neural Networks*, 2016.

[6] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell, "What do we need to build explainable AI systems for the medical domain?" no. January 2018, 2017. [Online]. Available: http://arxiv.org/abs/1712.09923

[7] R. Sinha and K. Swearingen, "The role of transparency in recommender systems," *Conference on Human Factors in Computing Systems - Proceedings*, 2002.

[8] J. L. Herlocker, J. A. Konstan, and J. Riedl, "Explaining collaborative filtering recommendations," *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 2000.

[9] P. L. Biecek, "Dalex: Explainers for complex predictive models in r," 2018.

[10] K. R. Varshney and H. Alemzadeh, "On the safety of machine learning: Cyber-physical sys-tems, decision sciences, and data products," *Big Data*, 2017.

[11] N. Foecking, M. Wang, and T. L. D. Huynh, "How do investors react to the data breaches news? empirical evidence from facebook inc. during the years 2016–2019," *Technology in Society*, 2021.

[12] Council of European Union, "Council regulation (EU) no 679/2016," 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj

[13] M. R. Carrillo, "Artificial intelligence: From ethics to law," *Telecommunications Policy*, 2020.

[14] "Beijing ai principles," *Datenschutz und Datensicherheit - DuD*, 2019. [Online]. Available: https://www.baai.ac.cn/news/beijing-ai-principles-en.html

[15] C. Röcker, "Chances and challenges of intelligent technologies in the production and retail sector," *World Acad. Sci. Eng. Technol.*, 2010.

[16] O. Korn, M. Funk, S. Abele, T. Hörz, and A. Schmidt, "Context-aware assistive systems at the workplace: Analyzing the effects of projection and gamification," in *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*, 2014.

[17] M. R. Endsley and D. B. Kaber, "Level of automation effects on performance, situation awareness and workload in a dynamic control task," *Ergonomics*, 1999.

[18] V. Lindström, M. Winroth, and J. Stahre, "Levels of automation in manufacturing 85 publications 187 citations see profile," 2008. [Online]. Available: http://www.researchgate.net/publication/255793362

[19] V. Panicker. (2020) Production management (me 3105). http://www.nitc.ac.in/app/webroot/img/upload/Production%20Management%20Module%201%20Course%20notes.pdf.

[20] x. Yang and D. A. Plewe. (2016) Assistance systems in manufacturing:a systematic review.

[21] A. A. Letichevsky, O. O. Letychevskyi, V. G. Skobelev, and V. A. Volkov, "Cybernetics cyber-physical systems," *Cybernetics and Systems Analysis*, vol. 53, 2017.

[22] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.

[23] X. Zheng, M. Wang, and J. Ordieres-Meré, "Comparison of data preprocessing approaches for applying deep learning to human activity recognition in the context of industry 4.0," *Sensors (Switzerland)*, vol. 18, no. 7, 2018.

[24] V. Michalis, N. Christophoros, and K. I. A., "A review of human activity recognition methods," 2015.

[25] A. Roitberg, A. Perzylo, N. Somani, M. Giuliani, M. Rickert, and A. Knoll, "Human activity recognition in the context of industrial human-robot interaction," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, 2014, pp. 1–10.

[26] T. Wenjin, L. M. C., and Y. Zhaozheng, "Multi-Modal Recognition of Worker Activity for Human-Centered Intelligent Manufacturing," aug 2019. [Online]. Available: http://arxiv.org/abs/1908.07519

[27] "Human Activity Recognition using Deep and Machine Learning Algorithms," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 4, 2020.

[28] M. Kok, J. Hol, and T. Schön, "Using inertial sensors for position and orientation estimation," *Foundations and Trends® in Signal Processing*, 2017.

[29] K. Y. Fang and V. Ragupathy, "Human and Machine Learning," *Computational Economics*, vol. 57, no. 3, pp. 889–909, mar 2021.

[30] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, and J. Peter Campbell, "Introduction to machine learning, neural networks, and deep learning," *Translational Vision Science and Technology*, vol. 9, no. 2, 2020.

[31] "Automated machine learning: Review of the state-of-the-art and opportunities for healthcare," apr 2020.

[32] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY: Springer, 2008.

[33] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning - wih Applications in R.* New York, NY: Springer, 2013.

[34] I. Salian. (2018) What's the difference between supervised, unsupervised, semi-supervised and reinforcement learning? [Online]. Available: https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/

[35] S. Shams. (2018) Evaluation of learning algorithm. [Online]. Available: https://machinelearningmedium.com/2018/04/02/evaluation-of-learning-algorithm/

[36] J. Brownlee. (2016) Overfitting and underfitting with machine learning algorithms. [Online]. Available: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/

[37] A. Mishra, "Amazon Machine Learning," *Machine Learning in the AWS Cloud*, pp. 317–351, 2019.

[38] A. Bhande. (2018) What is underfitting and overfitting in machine learning and how to deal with it.", medium. [Online]. Available: https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76

[39] Machine learning with matlab. [Online]. Available: https://explore.mathworks.com/interactive-machine-learning-with-matlab/chapter-4-1036-1040SB.html

[40] (2021) Support vector machine (svm). [Online]. Available: https://es.mathworks.com/discovery/support-vector-machine.html

[41] J. Suykens and J. Vandewalle, "Least Square Support Vector Machine Classifiers," *Neural Processing Letters*, 1999.

[42] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, Z. Yu, and S. Member, "Sensor-Based Activity Recognition," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS*, vol. 42, no. 6, pp. 790–808, 2012.

[43] J. M. Keller and M. R. Gray, "A Fuzzy K-Nearest Neighbor Algorithm," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-15, no. 4, pp. 580–585, 1985.

[44] (2015) A beginner's guide to neural networks and deep learning. [Online]. Available: https://wiki.pathmind.com/neural-network

[45] (2018) Feedforward deep learning models. [Online]. Available: http://uc-r.github.io/feedforward_DNN

[46] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2014.09.003

[47] A. Zell, "Simulation neuronaler netze [simulation of neural networks]," *Addison-Wesley*, 1994.

[48] *A modern introduction to probability and statistics : understanding why and how.* Springer, 1946.

[49] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[50] P. Ganesh. (2019) Types of convolution kernels. [Online]. Available: https://towardsdatascience.com/types-of-convolution-kernels-simplified-f040cb307c37

[51] cecbur. (2019) Convolutional neural network. [Online]. Available: https://commons.wikimedia.org/wiki/File:Convolutional_Neural_Network_NeuralNetworkFeatureLayers.gif

[52] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors (Switzerland)*, vol. 16, no. 1, jan 2016.

[53] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput*, vol. 9, p. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[54] E. Kang. (2017) Long short-term memory (lstm): Concept. [Online]. Available: https://medium.com/@kangeugine/long-short-term-memory-lstm-concept-cb3283934359

[55] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics (Switzerland)*, vol. 8, pp. 1–34, 2019.

[56] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," pp. 1–13, 2017. [Online]. Available: http://arxiv.org/abs/1702.08608

[57] C. Molnar, "Interpretable machine learning. a guide for making black boxmodels explainable." *Hands-On Machine Learning with R*, 2021.

[58] M. Pazzani, "Knowledge discovery from data?" *IEEE Intelligent Systems and their Applications*, vol. 15, no. 2, pp. 10–12, 2000.

[59] (2017) Machine learning: The power and promise of computers that learn by example. [Online]. Available: https://royalsociety.org/topics-policy/projects/machine-learning/

[60] (2018) Artificial intelligence for europe. [Online]. Available: https://ec.europa.eu/digital-single-market/en/news/communication-artificial-intelligence-europe

[61] (2018) Responsible ai practices—interpretability. [Online]. Available: https://ai.google/education/responsible-ai-practices?category=interpretability

[62] R. Schmelzer. (2019) Understanding explainable ai. [Online]. Available: https://www.forbes.com/sites/cognitiveworld/2019/07/23/understanding-explainable-ai/#28c157ca7c9e

[63] (2020) An introduction to xai. [Online]. Available: https://www.excella.com/wp-content/uploads/2020/04/Excella_XAI-eBook.pdf

[64] Giri. (2021) Explainable ai: What it is and why it matters. [Online]. Available: https://highdemandskills.com/explainable-ai/

[65] C. O'Sullivan. (2020) Interpretable vs explainable machine learning. [Online]. Available: https://towardsdatascience.com/interperable-vs-explainable-machine-learning-1fa525e12f48

[66] J. M. Alonso, "Teaching explainable artificial intelligence to high school students," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 974–987, 2020.

[67] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," 2018.

[68] P. Jonathon Phillips, C. A. Hahn, P. C. Fontana, and D. A. Broniatowski, "Draft NISTIR 8312 - Four Principles of Explainable Artificial Intelligence," 2020. [Online]. Available: https://doi.org/10.6028/NIST.IR.8312-draft

[69] D. Choujaa and N. Dulay, "Activity Recognition from Mobile Phone Data: State of the Art, Prospects and Open Problems," *Imperial College London*, vol. V, pp. 1–32, 2009. [Online]. Available: www.cityware.org.uk

[70] W. S. Lima, E. Souto, K. El-Khatib, R. Jalali, and J. Gama, "Human activity recognition using inertial sensors in a smartphone: An overview," *Sensors (Switzerland)*, vol. 19, no. 14, pp. 14–16, 2019.

[71] S. Rasnayaka and T. Sim, "Your Tattletale Gait Privacy Invasiveness of IMU Gait Data," *IJCB 2020 - IEEE/IAPR International Joint Conference on Biometrics*, no. March, 2020.

[72] M. C. Mozer, "The neural network house: An environment that adapts to its inhabitants," in *in Proc. AAAI Spring Symp. Intell. Environ.*

[73] P. Rivera, E. Valarezo, M.-T. Choi, and T.-S. Kim, "Recognition of Human Hand Activities Based on a Single Wrist IMU Using Recurrent Neural Networks," *International Journal of Pharma Medicine and Biological Sciences*, vol. 6, no. 4, pp. 114–118, 2017. [Online]. Available: http://www.ijpmbs.com/uploadfile/2017/1227/20171227050020234.pdf

[74] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Commun*, vol. 48, pp. 140–150, 2010.

[75] T. Sohn, A. Varshavsky, A. Lamarca, M. Y. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. G. Griswold, and E. D. Lara, "Mobility detection using everyday gsm traces," in *In International Conference on Ubiquitous Computing*.

[76] I. Anderson, J. Maitland, S. Sherwood, L. Barkhuus, M. Chalmers, M. Hall, B. Brown, and H. Muller, "Shakra: Tracking and sharing daily activity levels with unaugmented mobile phones," *Mob. Netw. Appl*, vol. 12, pp. 185—1998, 2007. [Online]. Available: http://www.ijpmbs.com/uploadfile/2017/1227/20171227050020234.pdf

[77] S. Dernbach, B. Das, N. Krishnan, B. Thomas, and D. Cook, "Simple and complex activity recognition through smart phones," in *In Proceedings of the 2012 Eighth International Conference on Intelligent Environments, Guanajuato, Mexico*.

[78] O. Banos, J. M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition." *Sensors*, 2014.

[79] A. W. Kempa-Liehr, J. Oram, A. Wong, M. Finch, and T. Besier, "Feature Engineering Workflow for Activity Recognition from Synchronized Inertial Measurement

Units," *Communications in Computer and Information Science*, vol. 1180 CCIS, pp. 223–231, 2020.

[80] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Personal and Ubiquitous Computing*, vol. 14, no. 7, pp. 645–662, 2010.

[81] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," in *in Proc. Pervasive*.

[82] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *in Proc. 17th Conf. Innovative Appl. Artif. Intell.*

[83] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," *International Workshop of Ambient Assisted Living (IWAAL 2012)*, 2012.

[84] N. Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables," apr 2016. [Online]. Available: http://arxiv.org/abs/1604.08880

[85] I. Anderson and H. Muller, "Context awareness via gsm signal strength fluctuation," in *InPervasive '06: Proceedings of the 4th International Conference on Pervasive Computing*.

[86] S. Ray Hong, J. Hullman, and E. Bertini, "Human factors in model interpretability: Industry practices, challenges, and needs," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, pp. 1–27, 2020.

[87] W. Lima, E. Souto, T. Rocha, R. Pazzi, and F. Pramudianto, "User activity recognition for energy saving in smart home environment," in *In Proceedings of the IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus.*

[88] J. L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 1 2016.

[89] L. Köping, K. Shirahama, and M. Grzegorzek, "A general framework for sensor-based human activity recognition," *Comput. Biol. Med.*, 2018.

[90] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," 10 2019. [Online]. Available: http://arxiv.org/abs/1910.10045

[91] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," vol. 13-17-August-2016. Association for Computing Machinery, 8 2016, pp. 1135–1144.

[92] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 12 2013. [Online]. Available: http://arxiv.org/abs/1312.6034

[93] P. Parvatharaju, T. Hartvigsen, and E. Rundensteiner, "Learning Saliency Maps for Deep Time Series Classifiers," pp. 1406–1415, 2021.

[94] A. Schreiber. (2019) Saliency maps for deep learning: Vanilla gradient. [Online]. Available: https://andrewschrbr.medium.com/saliency-maps-for-deep-learning-part-1-vanilla-gradient-1d0665de3284

[95] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," 6 2018. [Online]. Available: http://arxiv.org/abs/1806.07421

[96] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 5 2017. [Online]. Available: http://arxiv.org/abs/1705.07874

[97] S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K. W. Low, S. F. Newman, J. Kim, and S. I. Lee, "Explainable machine learning predictions to help anesthesiologists prevent hypoxemia during surgery," *bioRxiv*, 2017.

[98] D. Sarkar. (2018) Model interpretation strategies. [Online]. Available: https://towardsdatascience.com/explainable-artificial-intelligence-part-2-model-interpretation-strategies-75d4afa6b739

[99] J. L. Reyes-Ortiz, D. Anguita, L. Oneto, and X. Parra. (2015) Smartphone-based recognition of human activities and postural transitions data set. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions

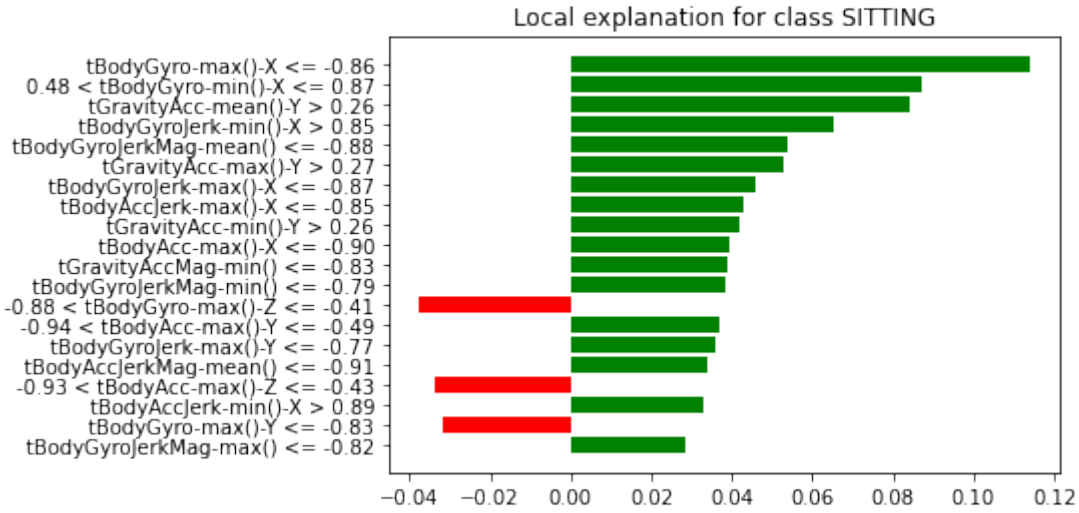
[100] M. Sepahvand and F. Abdali-Mohammadi, "A novel representation in genetic programming for ensemble classification of human motions based on inertial signals," *Expert Systems with Applications*, vol. 185, 12 2021.

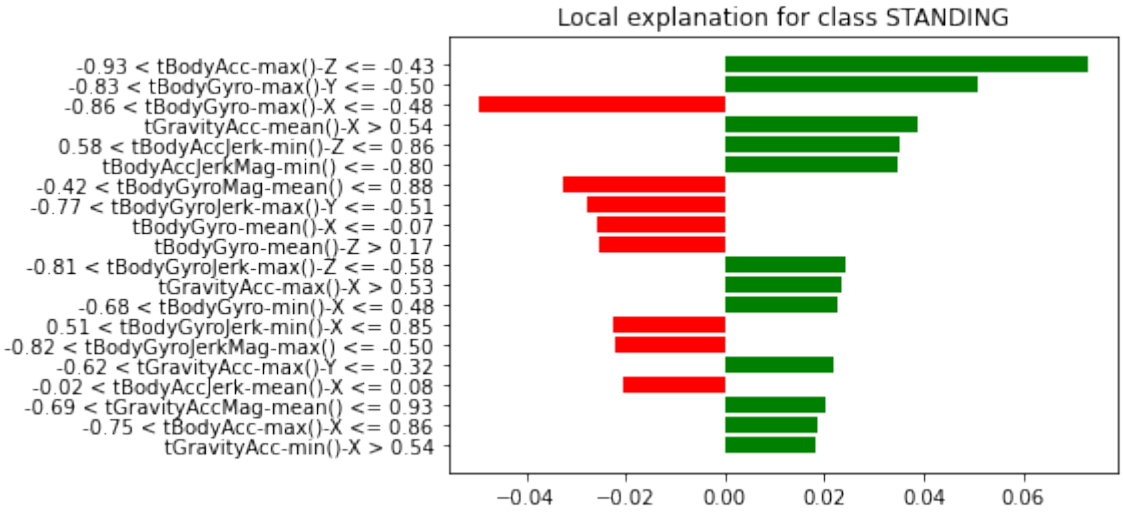
[101] R. Sunil. (2015) How to increase accuracy of machine learning model. [Online]. Available: https://www.analyticsvidhya.com/blog/2015/12/improve-machine-learning-results/

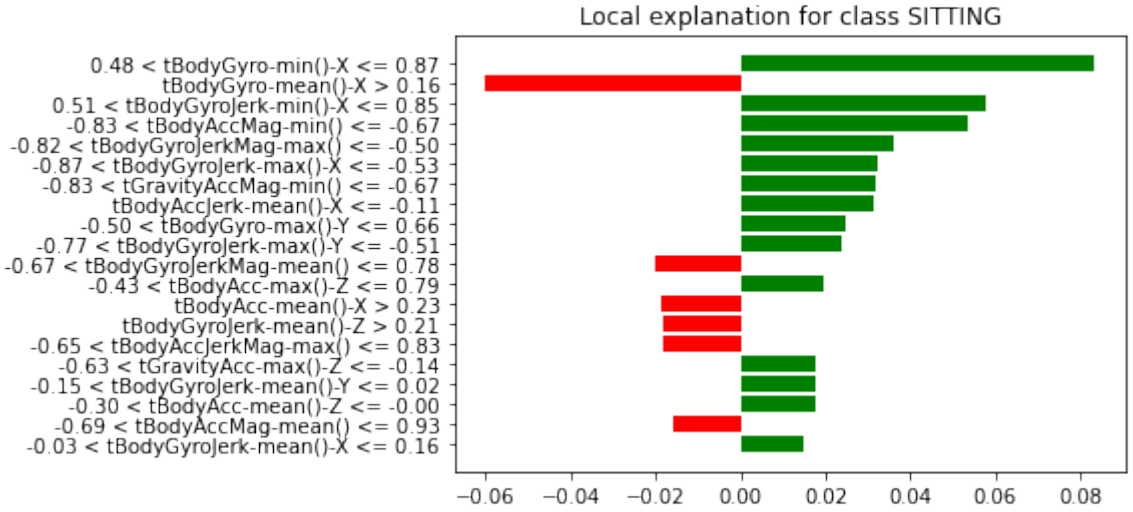
[102] Y. Zhang, Z. Zhang, Y. Zhang, J. Bao, Y. Zhang, and H. Deng, "Human activity recognition based on motion sensor using u-net," 2019.

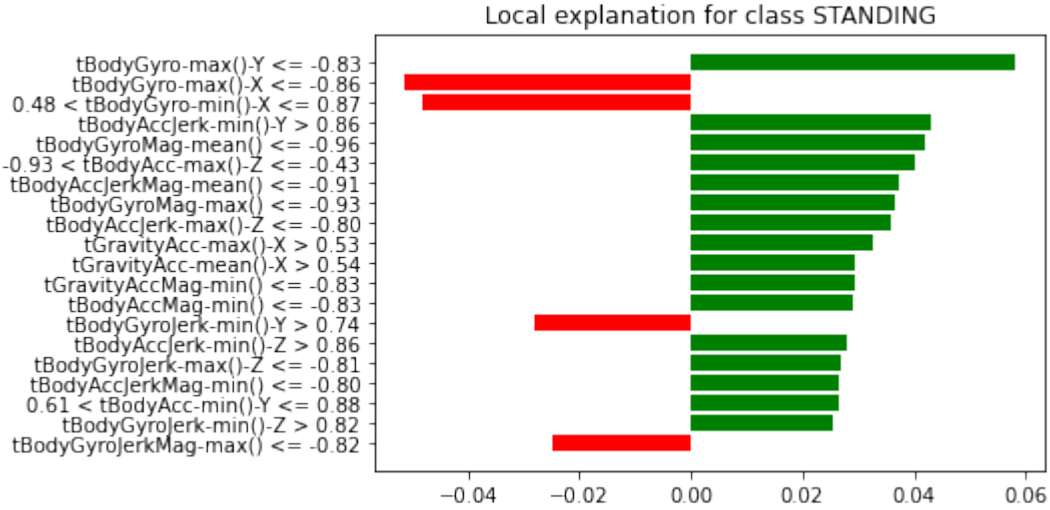
[103] Y. Ma and H. Ghasemzadeh, "Labelforest: Non-parametric semi-supervised learning for activity recognition," p. 19. [Online]. Available: https://github.com/y-max/LabelForest.

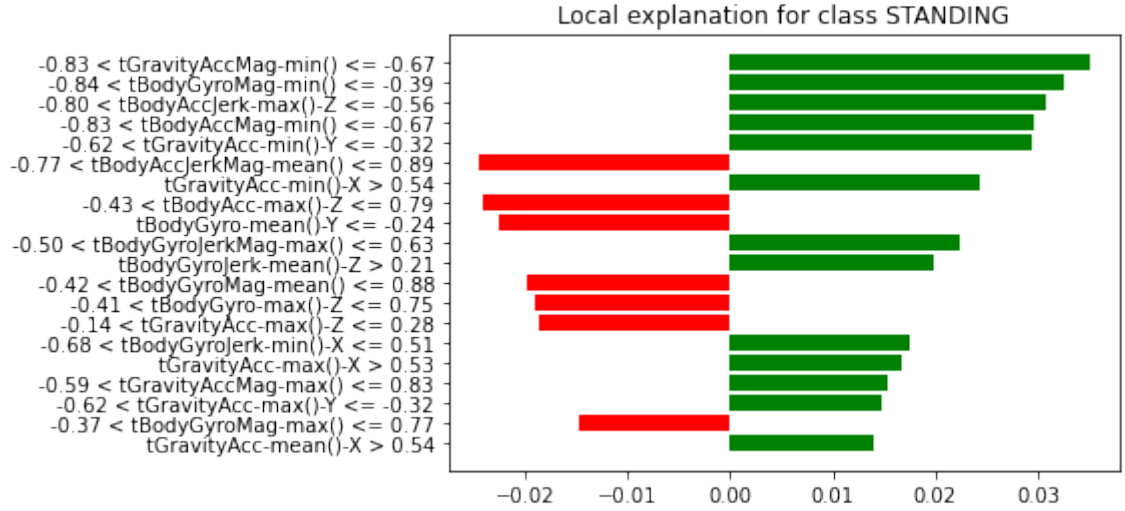
A. SP-LIME Results - Logistic Regression

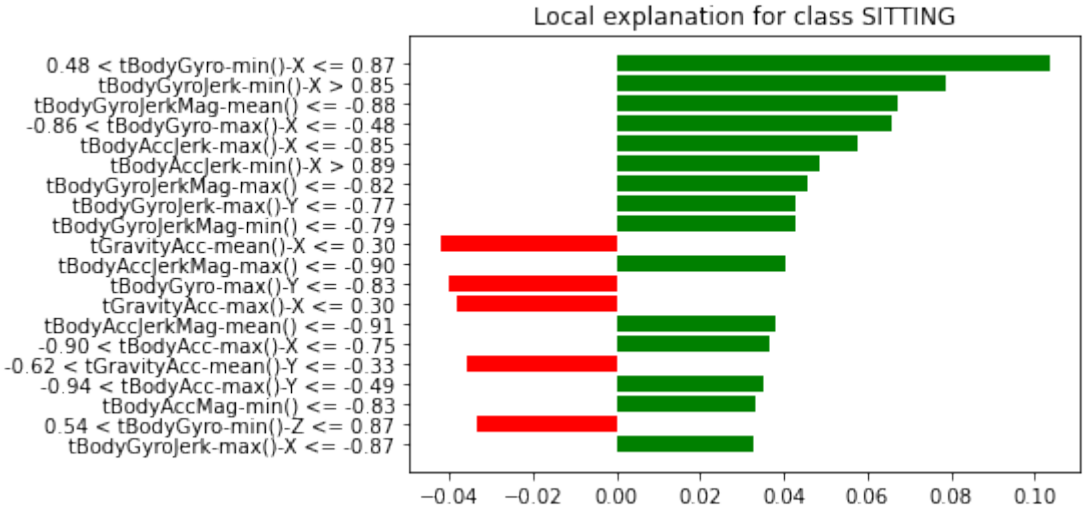


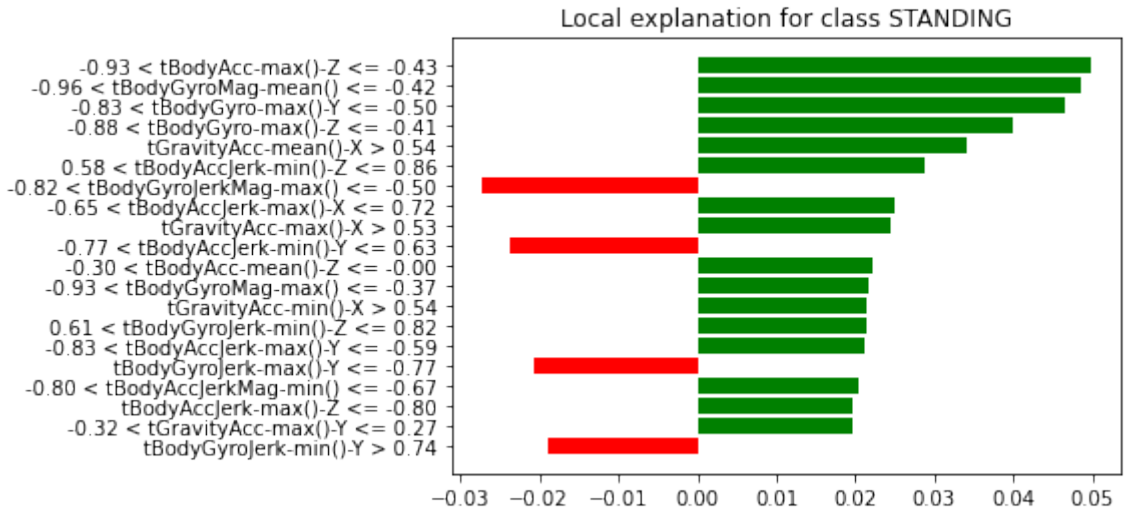


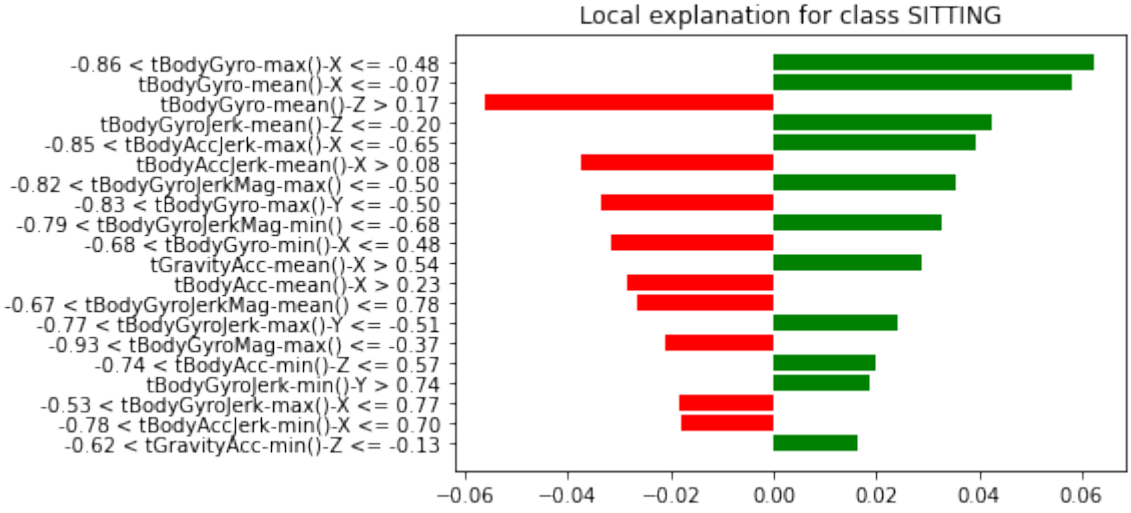


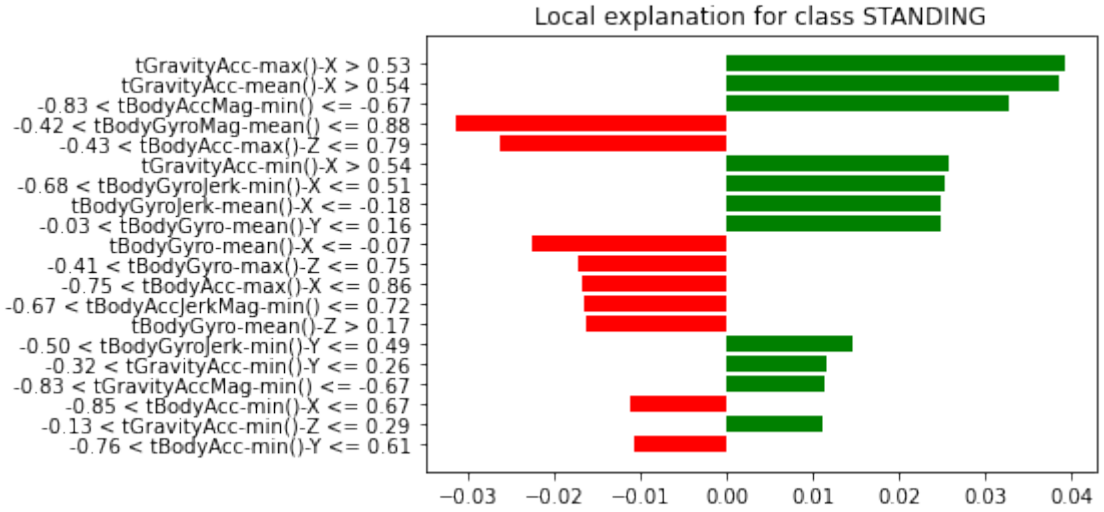


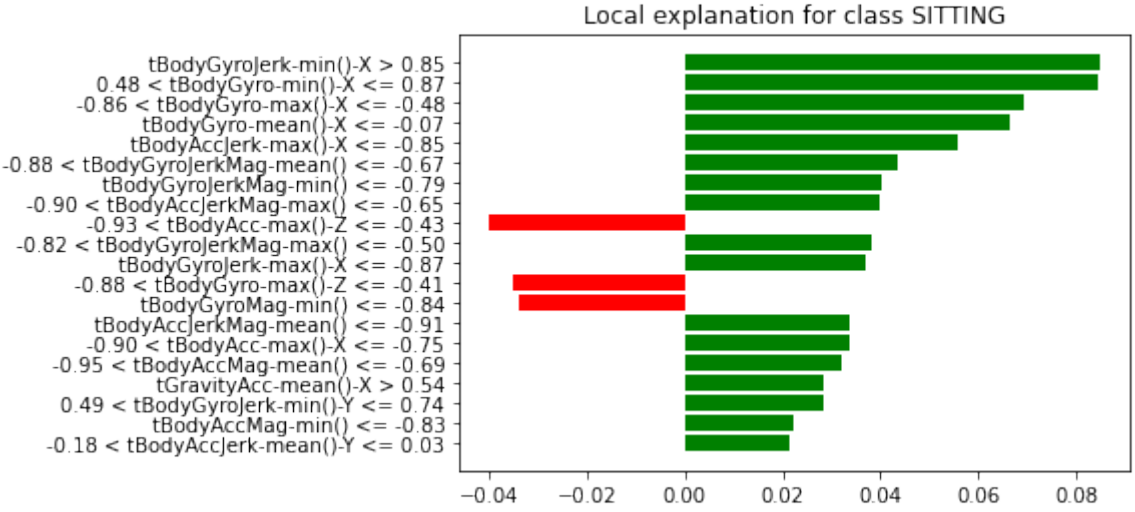


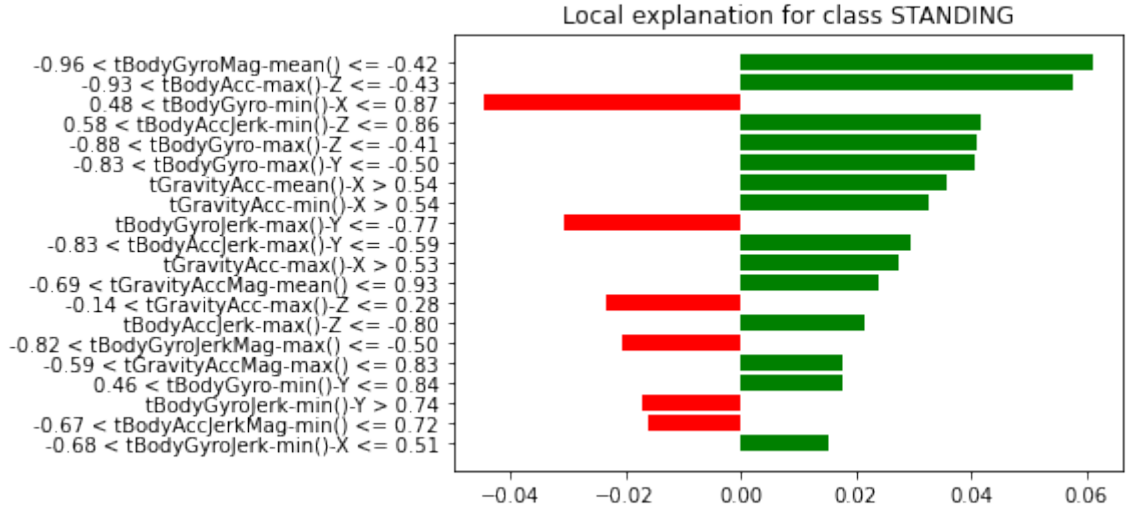












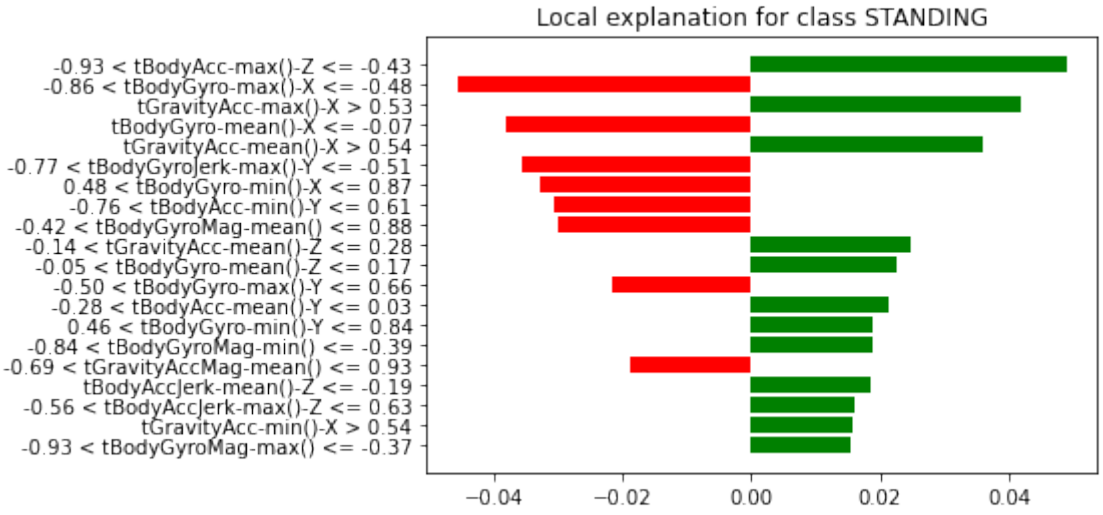
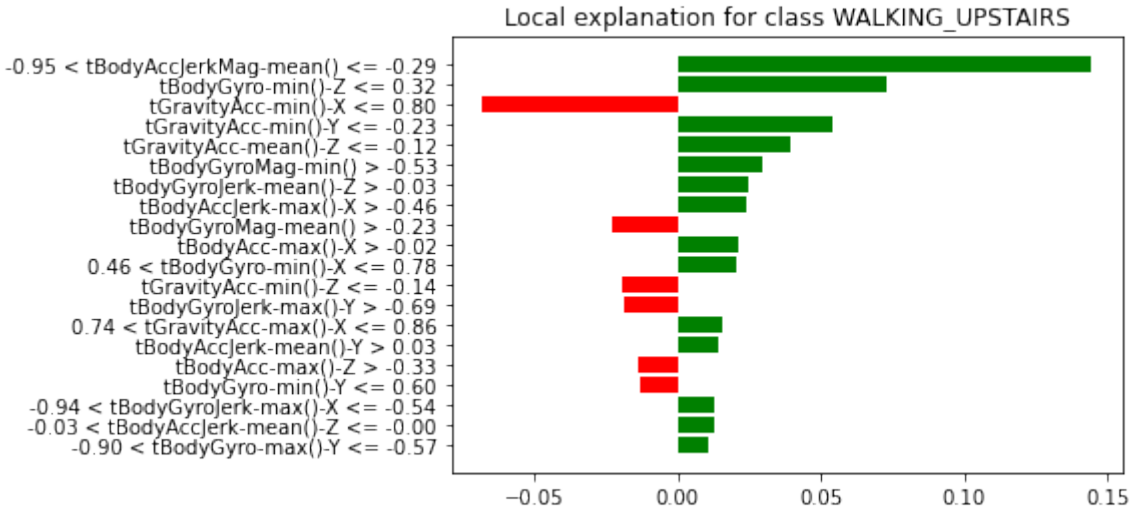
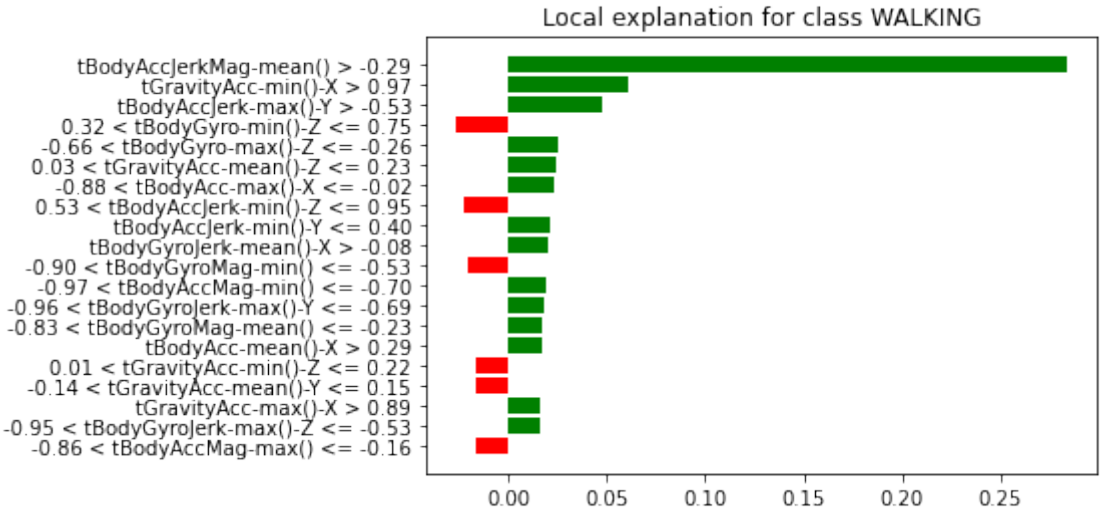
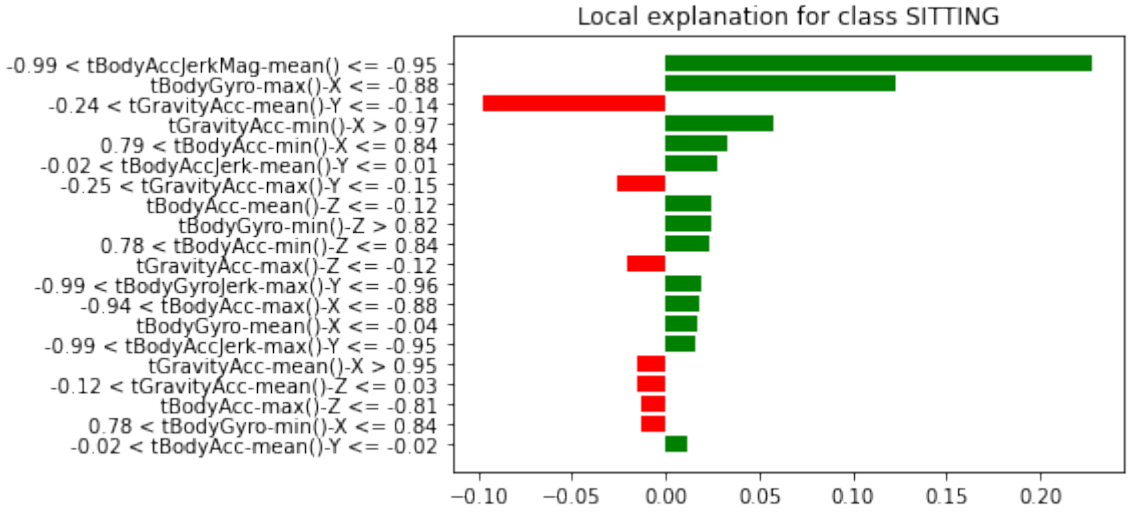


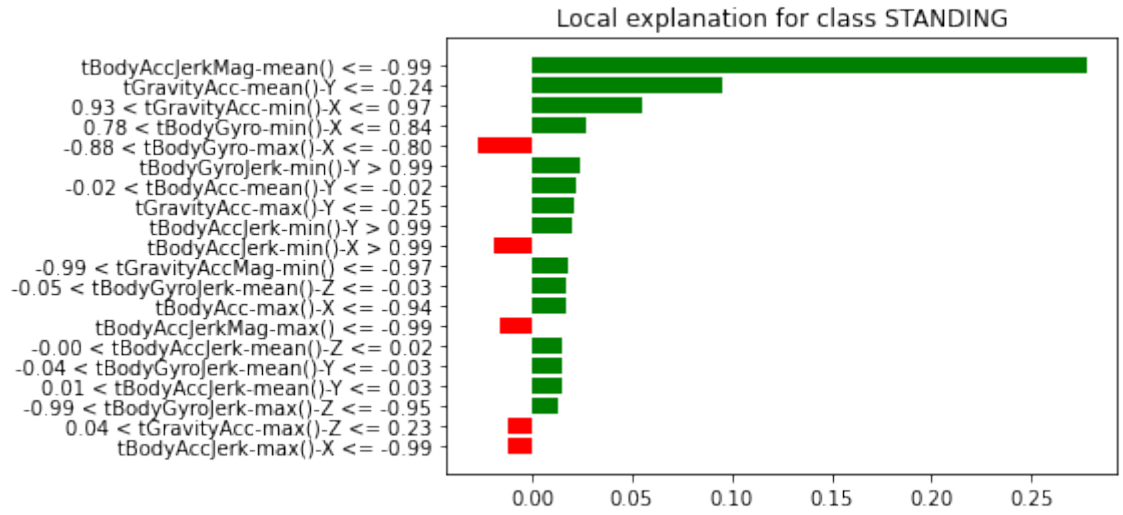
Figure A.1.: Local explanations for twelve significant instances provided by LIME and Submodular-Pick LIME for the Logistic Regression algorithm.

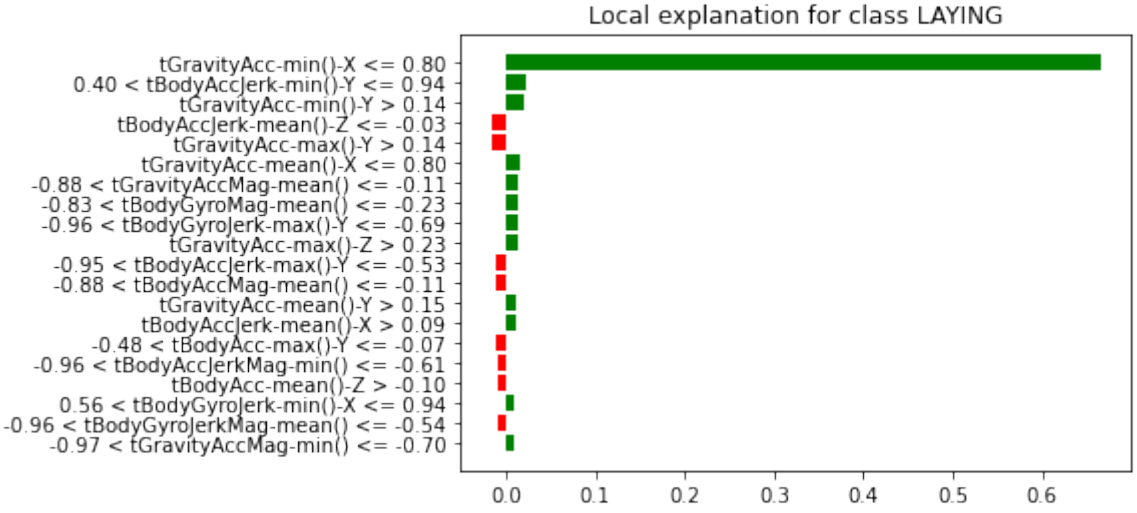
B. SP-LIME Results - Decision Tree

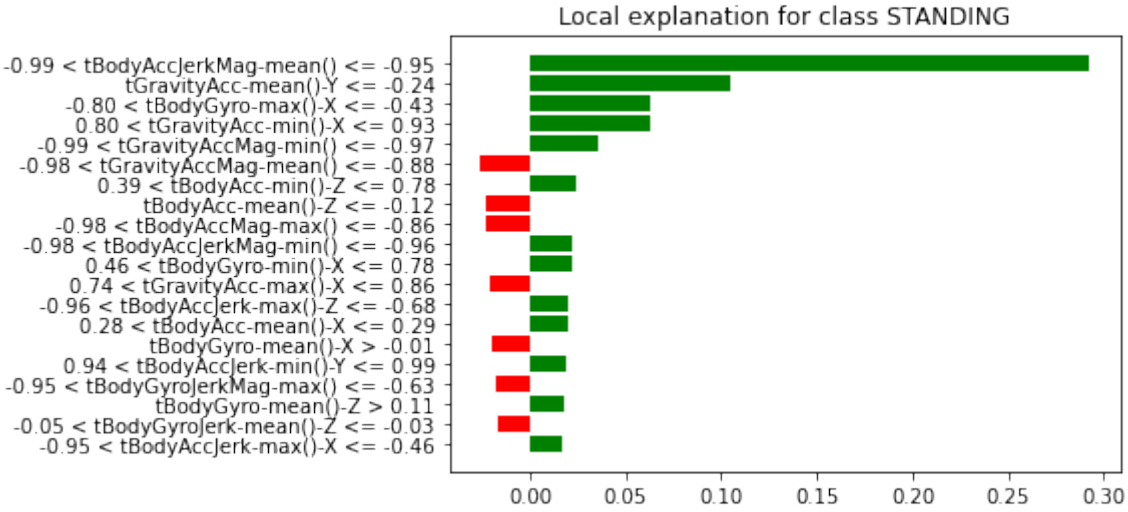


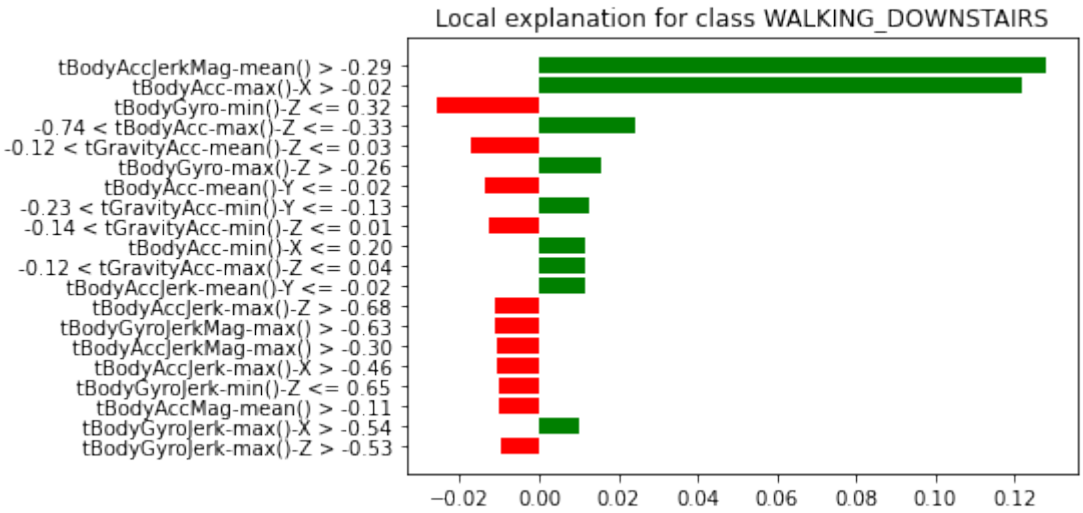


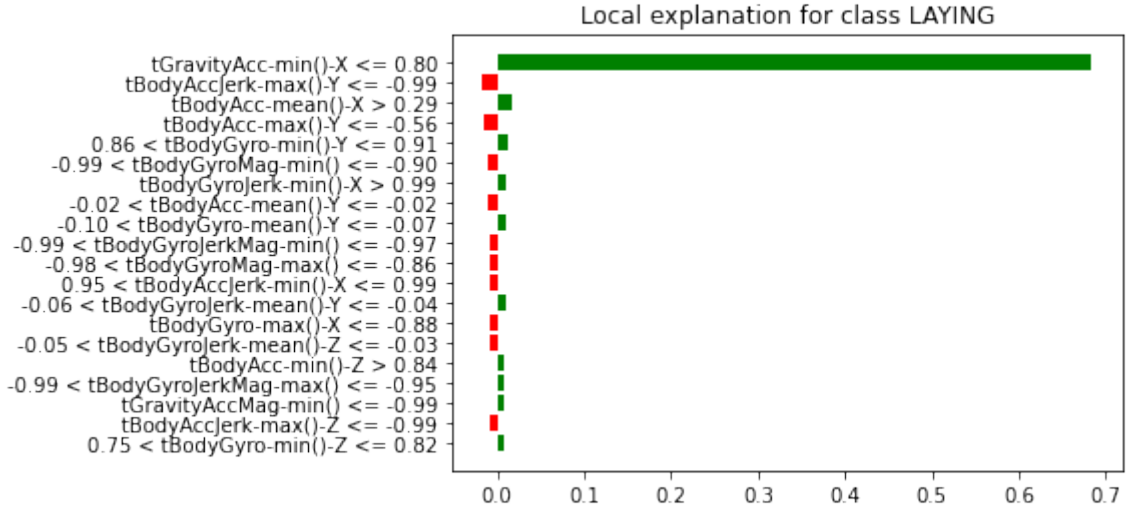


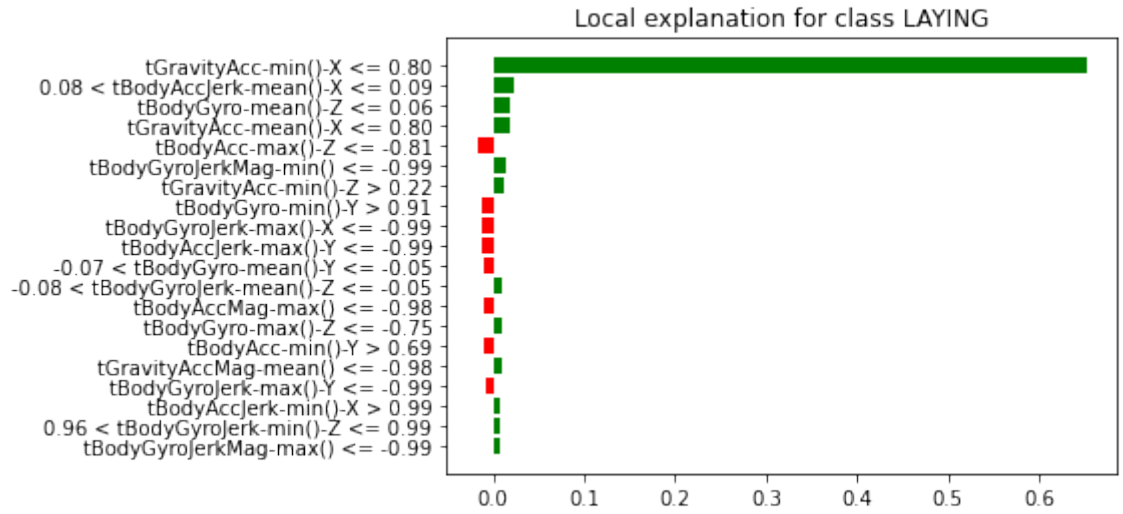


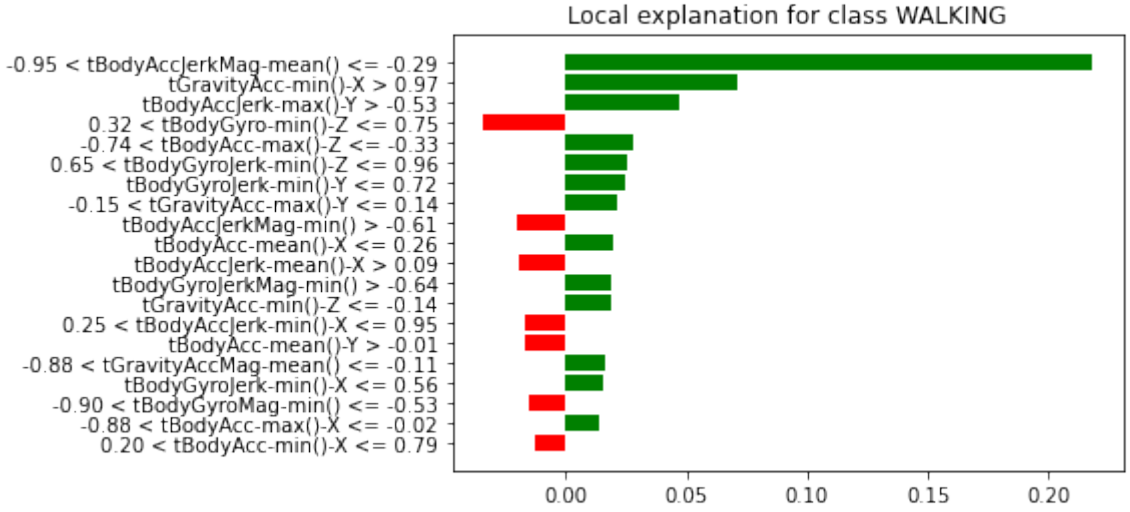


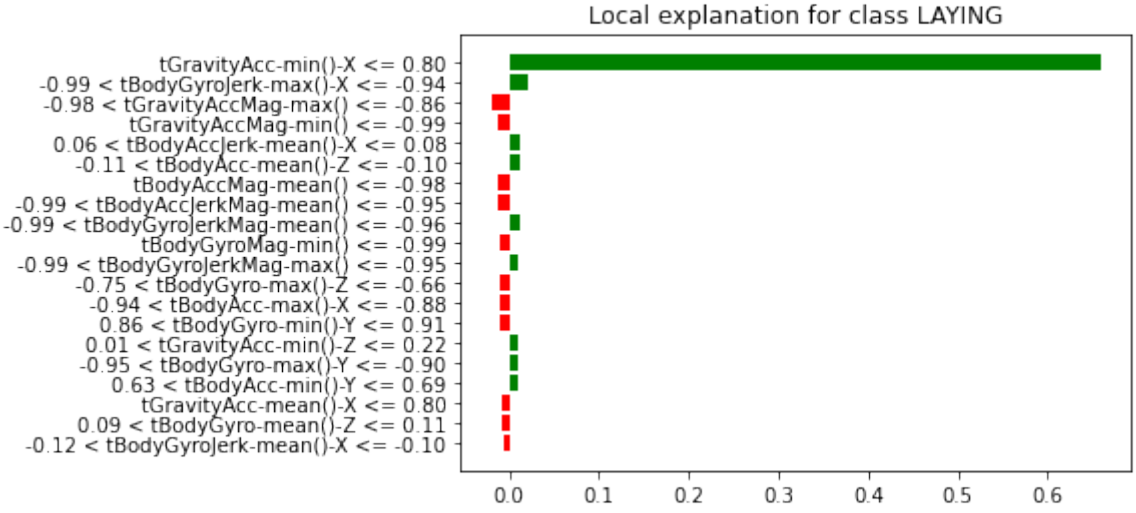












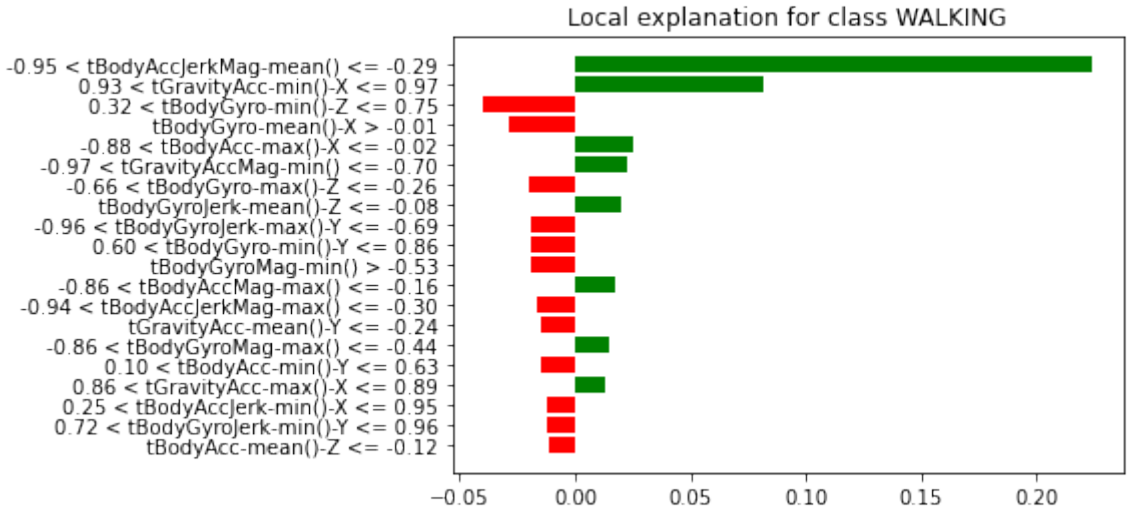
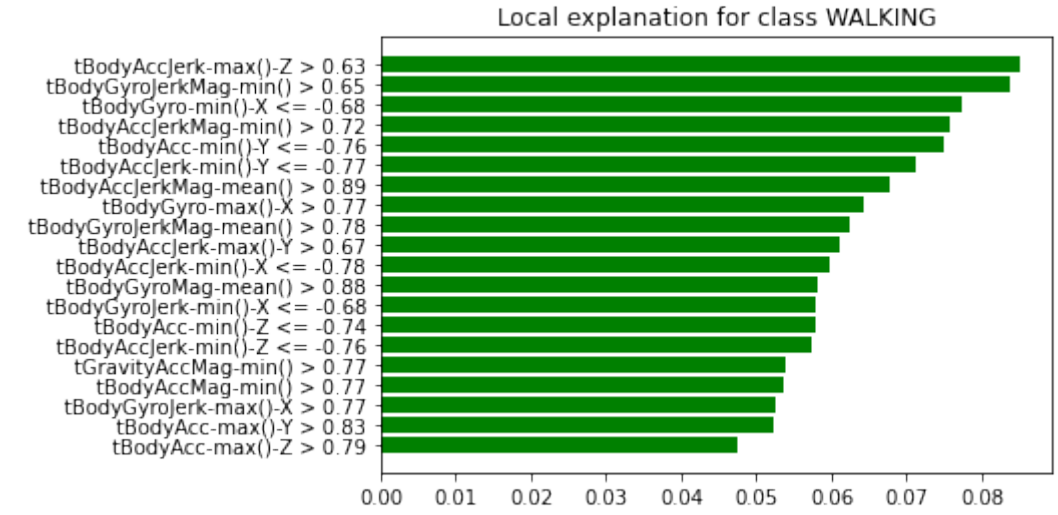
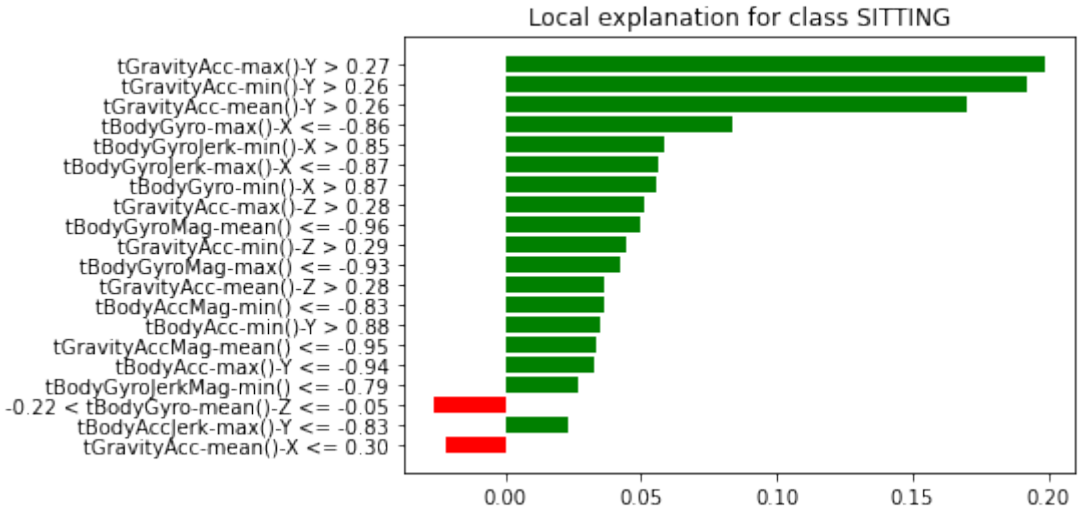
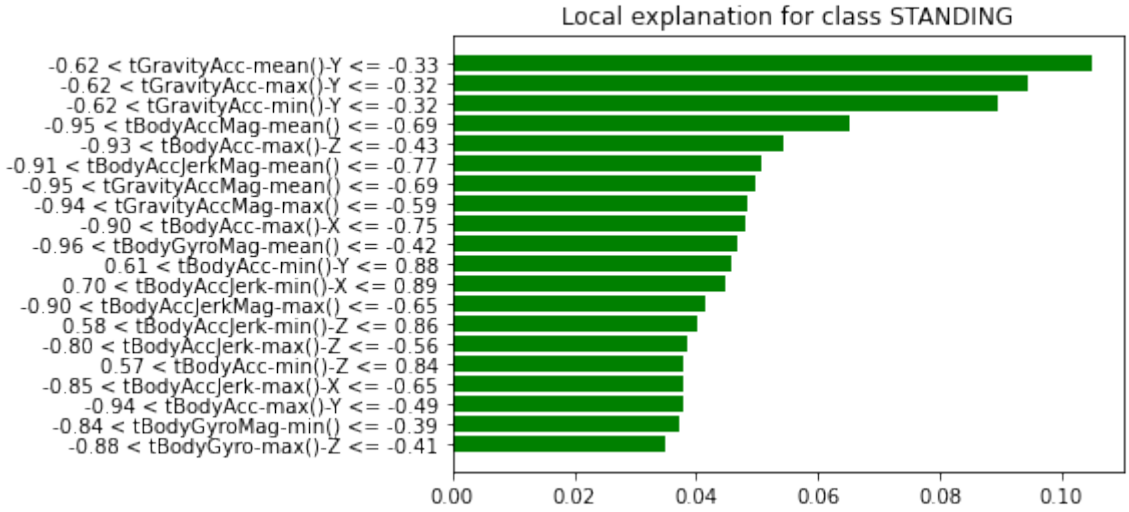


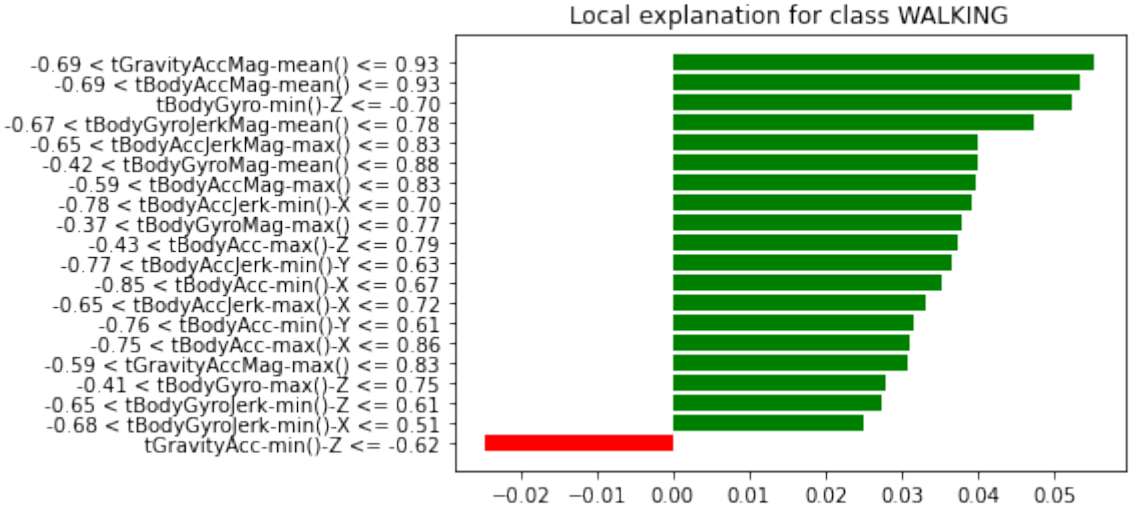
Figure B.1.: Local explanations for twelve significant instances provided by LIME and Submodular-Pick LIME for the Decision Tree algorithm.

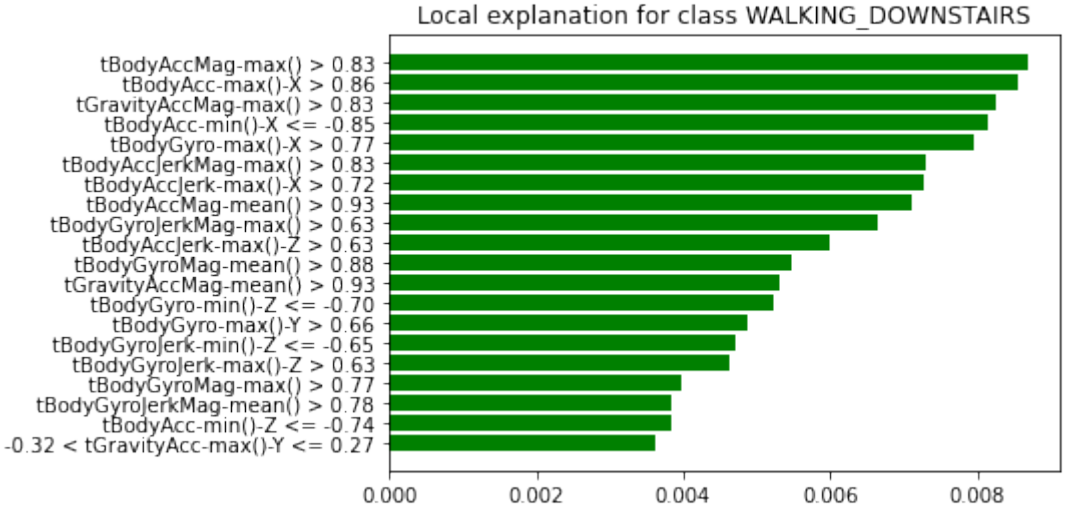
C. SP-LIME Results - K-Nearest Neighbour

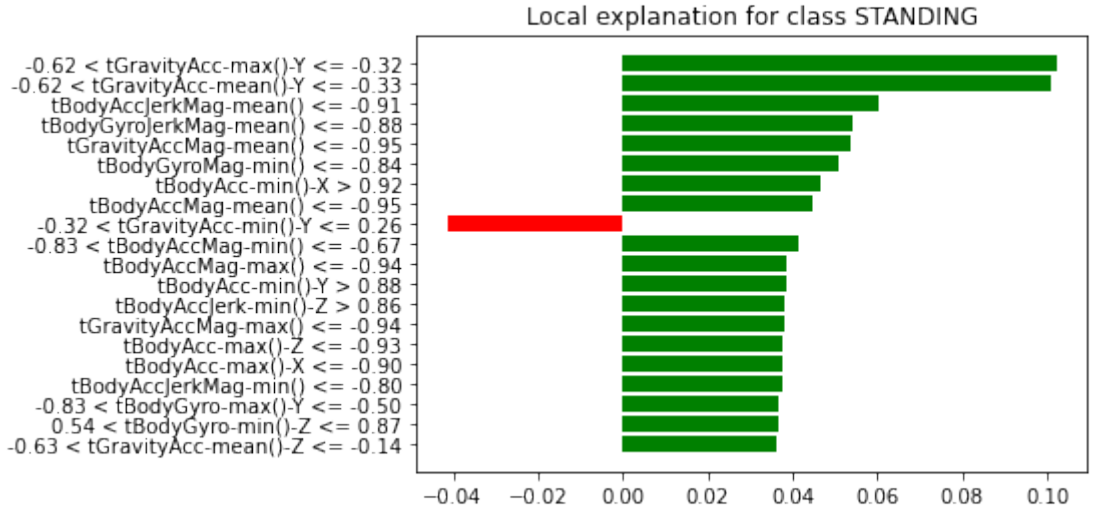


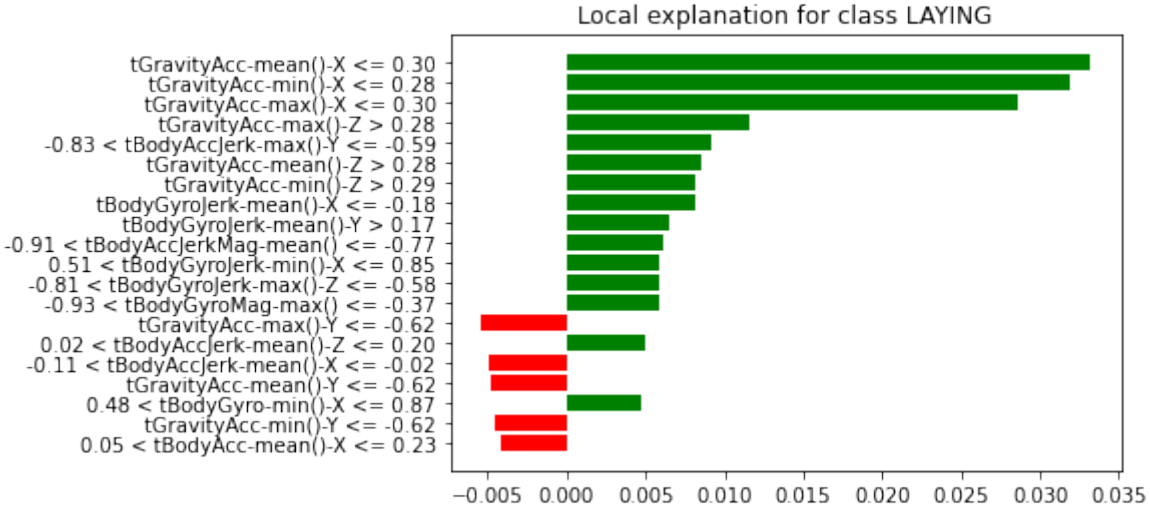


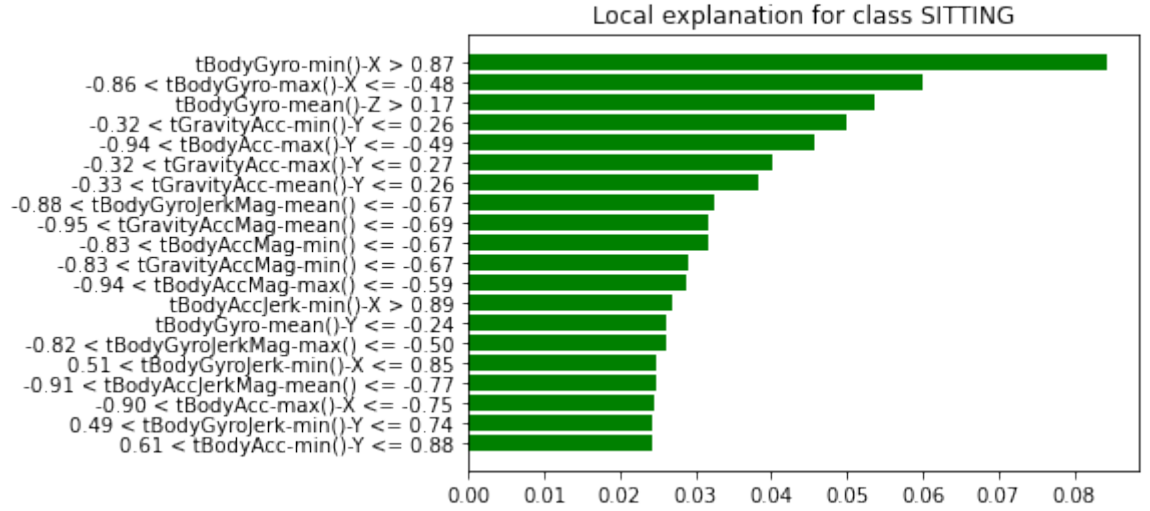


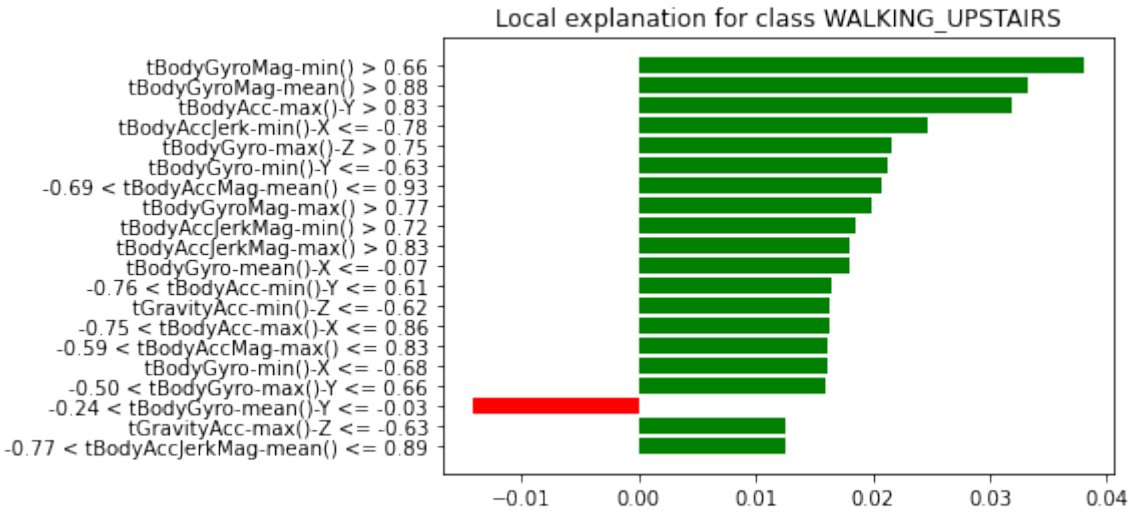


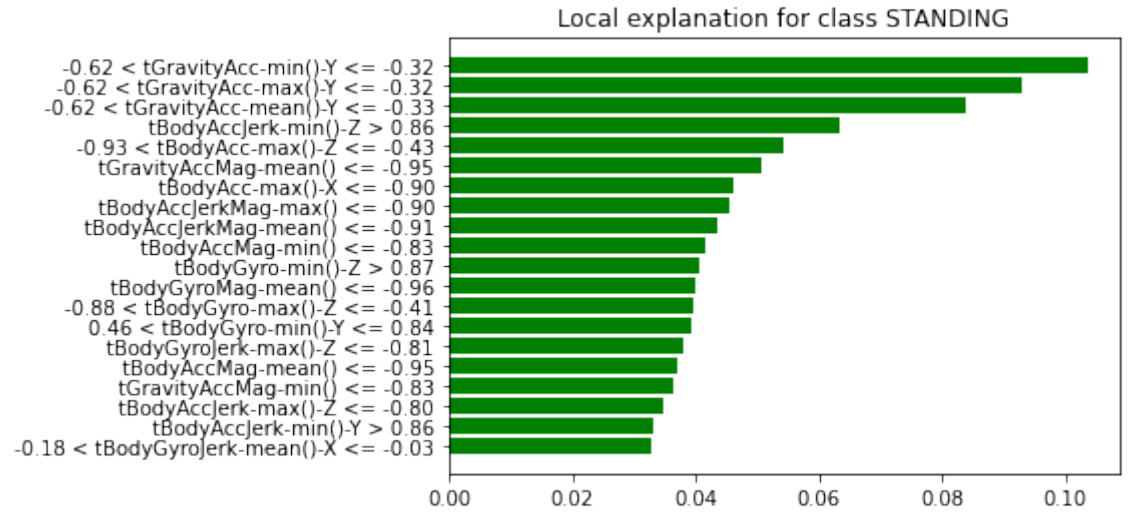


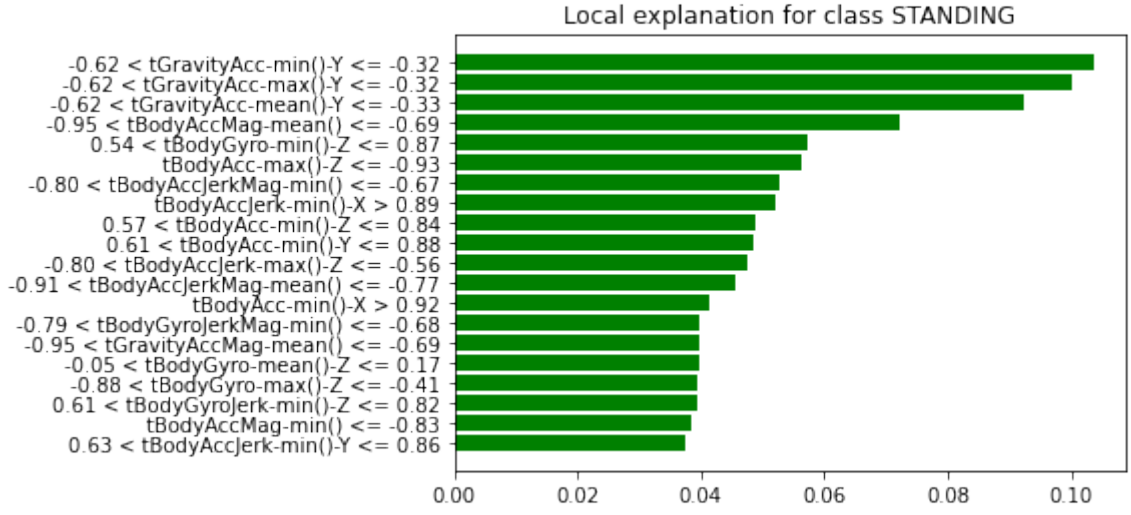












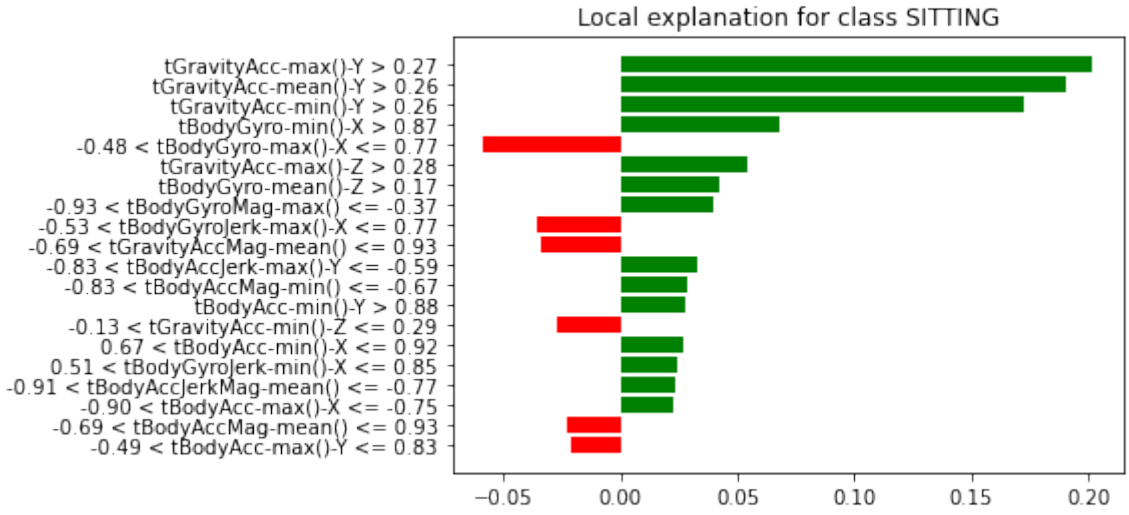
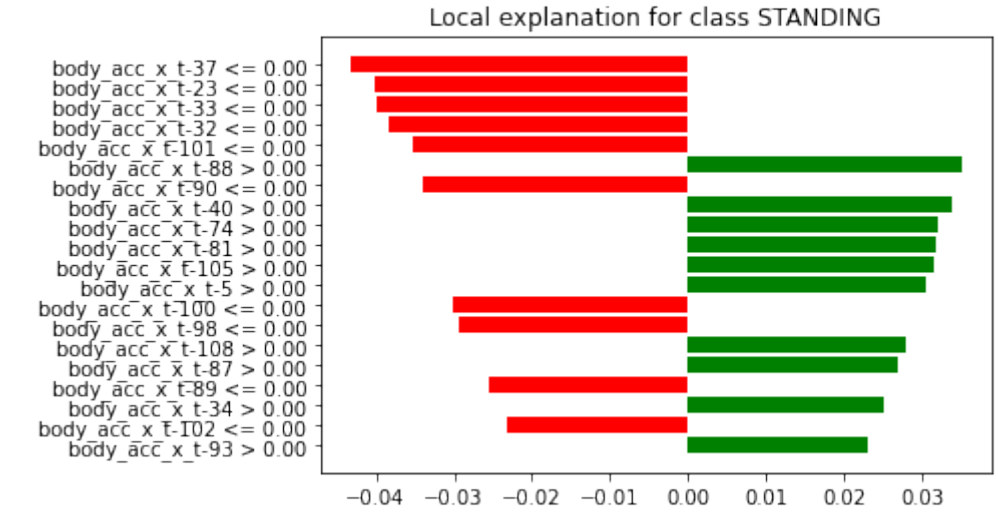
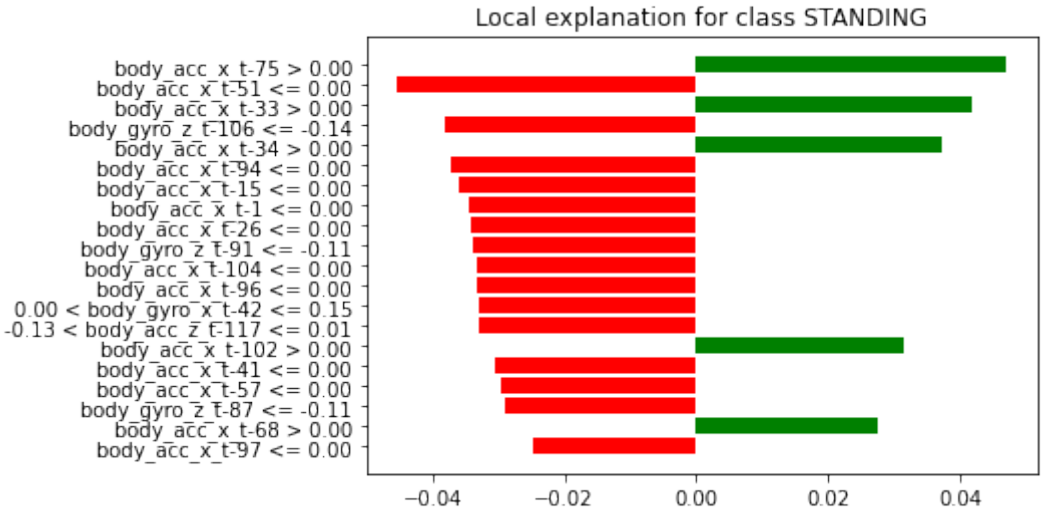
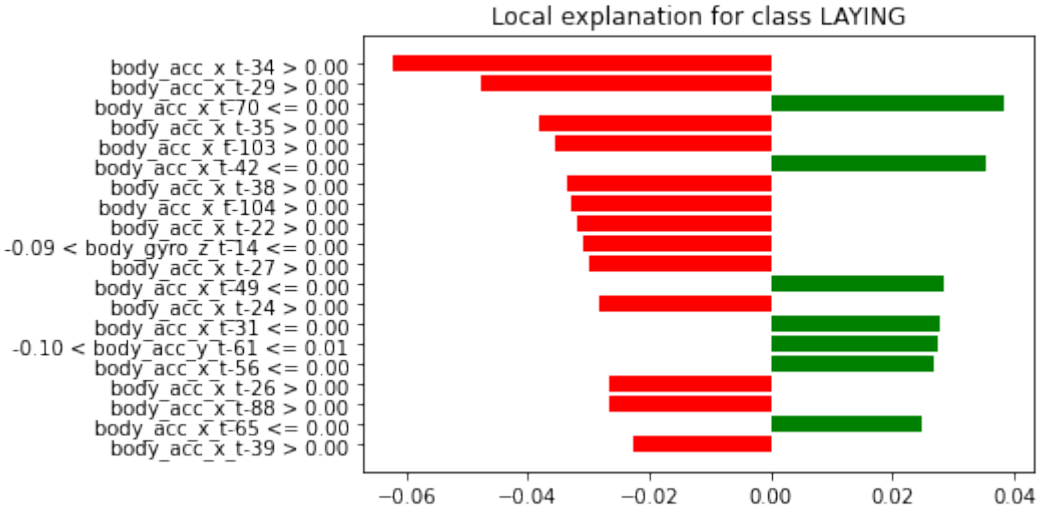


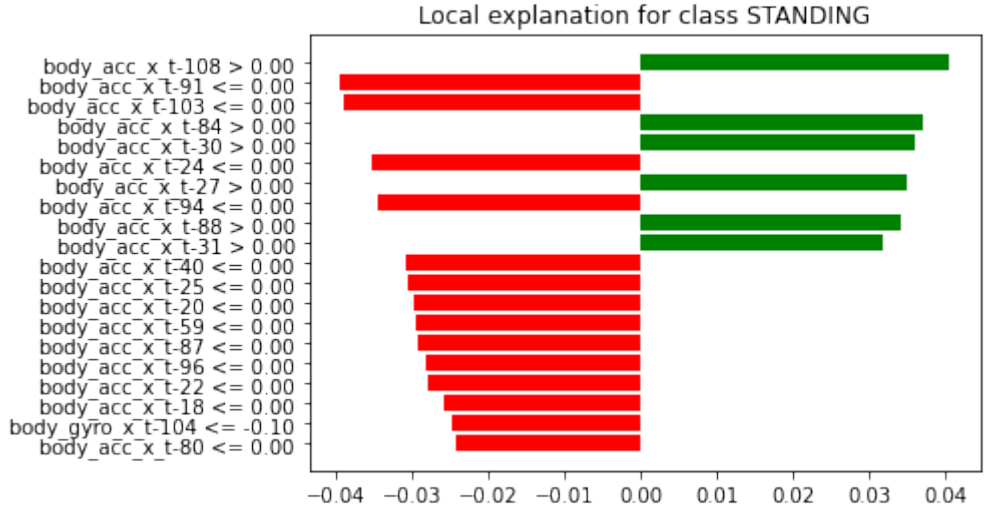
Figure C.1.: Local explanations for twelve significant instances provided by LIME and Submodular-Pick LIME for the k-nearest neighbour algorithm.

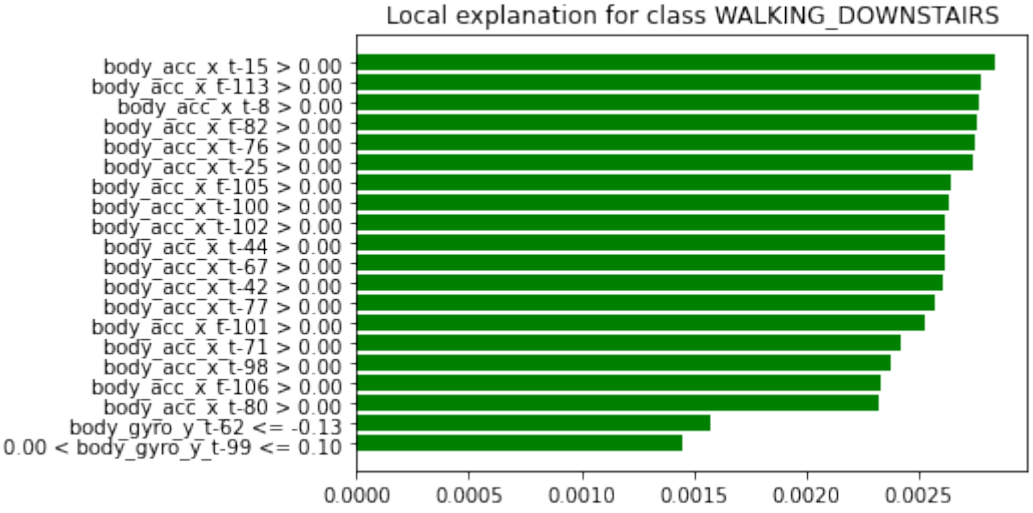
D. SP-LIME Results - CNN-LSTM Neural Network

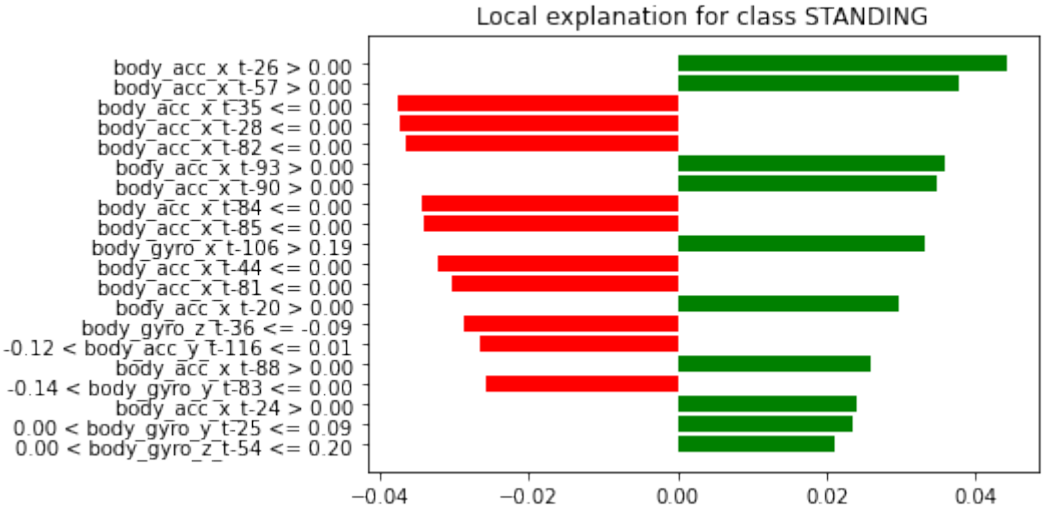


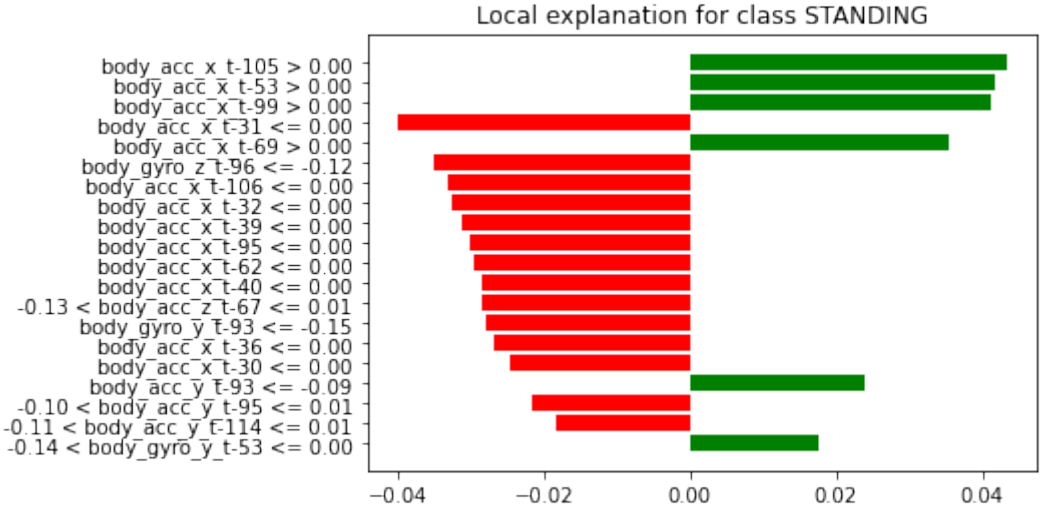


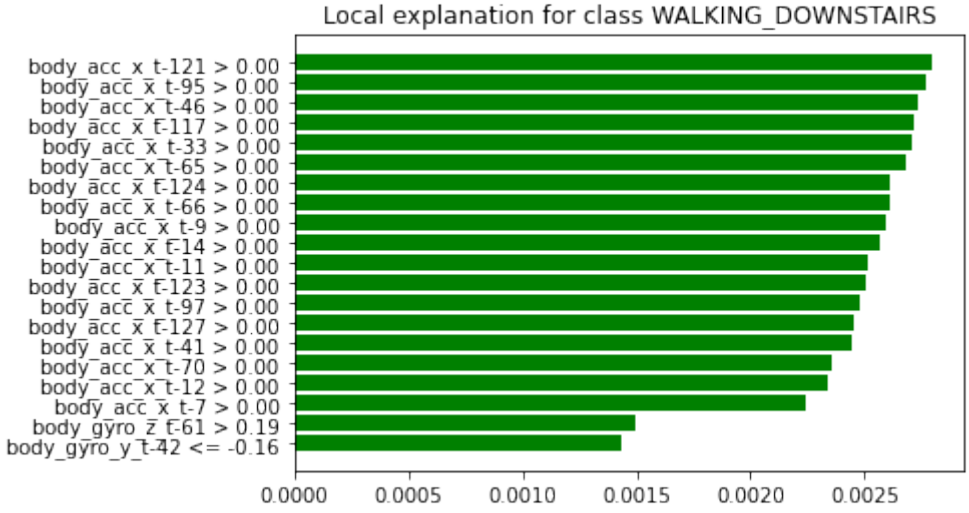


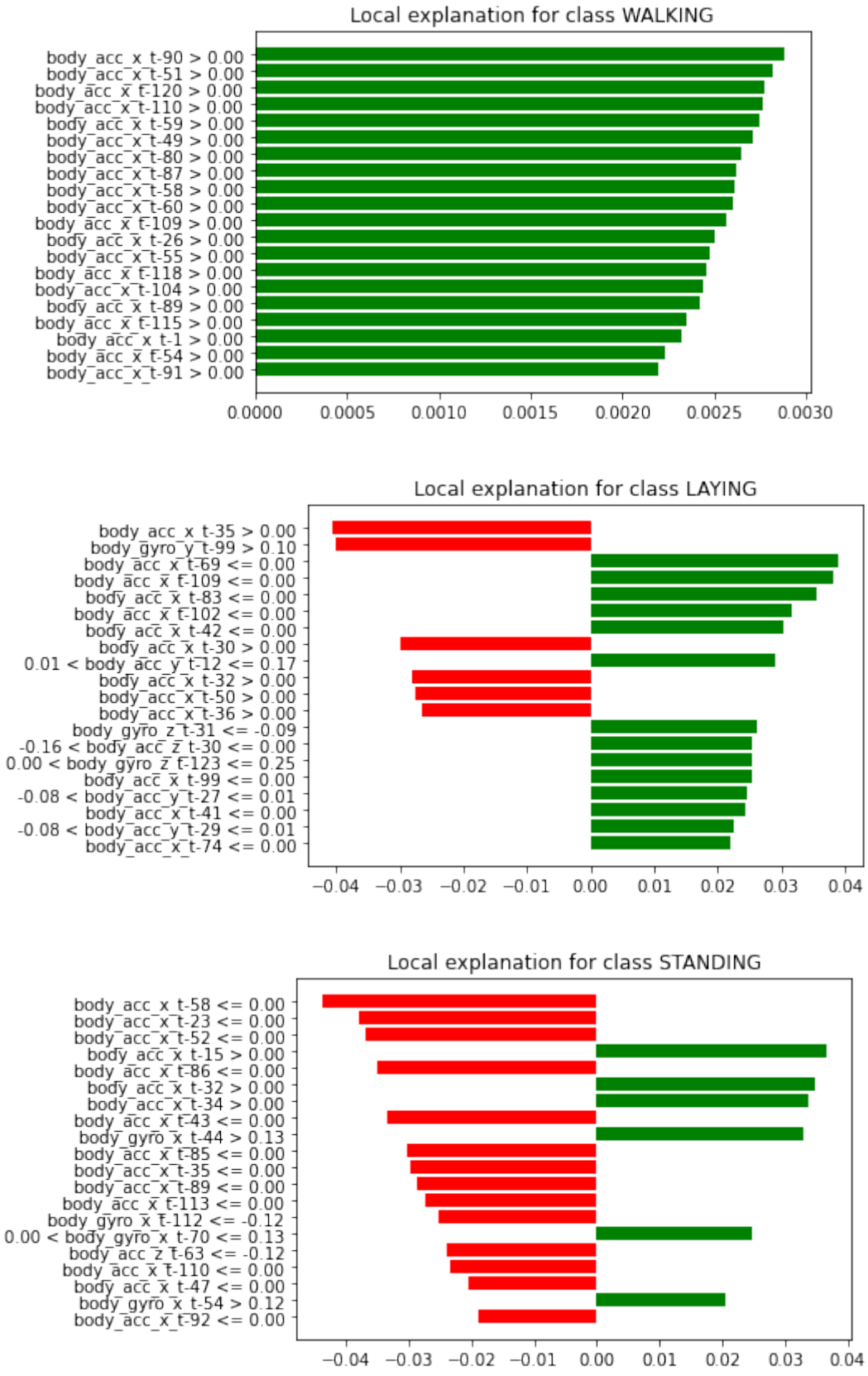












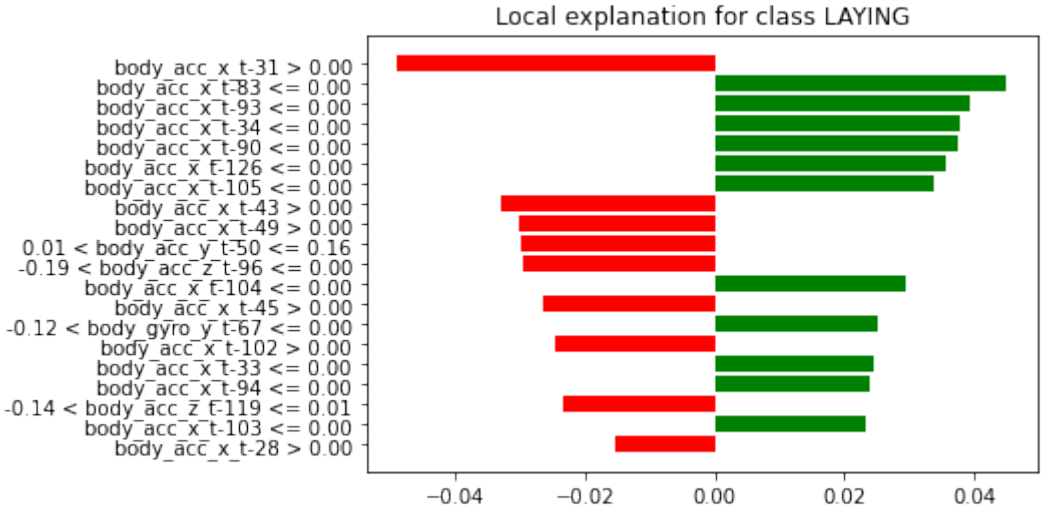


Figure D.1.: Local explanations for twelve significant instances provided by LIME and Submodular-Pick LIME for the convolutional and long short-termed memory neural network algorithm.