An Architecture for Predicting Fall Events in the Home of Elderly People

Gabriel Di iorio Silva

JUIZ DE FORA

JANEIRO, 2023

An Architecture for Predicting Fall Events in the Home of Elderly People

Gabriel Di iorio Silva

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Victor Ströele de Andrade Menezes

JUIZ DE FORA
JANEIRO, 2023

An Architecture for Predicting Fall Events in the Home of Elderly People

Gabriel Di iorio Silva

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Victor Ströele de Andrade Menezes
Phd - Universidade Federal do Rio de Janeiro

Mario Antônio Ribeiro Dantas
Phd - University of Southampton

Heder Soares Bernardino
Phd - Laboratório Nacional de Computação

JUIZ DE FORA

12 DE JANEIRO, 2023

To my parents that made all this possible with their effort in raising a child with the best education possible

To all my friends that I've made during university that, for sure, I'm gonna carry for my entire life.

Resumo

Com a Internet das Coisas se tornando mais popular ano após ano, não é incomum identificar sensores em uma grande variedade de objetos domésticos. As tecnologias para reduzir a escala dos sensores possibilitaram transformar uma casa em uma casa inteligente e ajudar pessoas com necessidades especiais. Dessa forma, os dados digitais são gerados massivamente, e inúmeros estudos e resultados são possíveis com essas entradas, como detecção e previsão de eventos. Portanto, este trabalho propõe uma arquitetura baseada em Fog-Cloud com teorias do paradigma Lambda para prever eventos de queda de idosos. Para atingir este objetivo, foram feitos experimentos com dados coletados de trabalhos anteriores sobre detecção de quedas já testados e validados. Assim, um passo a mais é dado ao analisar essas informações para compreender os melhores resultados possíveis considerando a sensibilidade e a janela de tempo para agir quando uma previsão de queda é realizada.

Abstract

With the Internet of Things becoming more popular year after year, it's not unusual to identify sensors in a large variety of home objects. Technologies to reduce the scale of sensors made it possible to transform a home into a smart home and help people with special needs. This way, digital data is generated massively, and numerous studies and outcomes are possible with these inputs, such as event detection and prediction. Therefore, this work proposes an architecture based on Fog-Cloud with theories from the Lambda paradigm to predict falling events of elderly people. To achieve this goal, experiments were made with data collected from previous work about falling detection already tested and validated. Thus, a step further is done through analyzing this information to comprehend the best results possible considering sensitiveness and the time window to act when a prediction of falling is performed.

Keywords: Time Series, Event Forecasting, Human Activity, Ambient Assisted Living, Elderly Falling, Binary Classification, Imbalanced Data.

Acknowledgements

To my parents and all my friends who made this possible with their love and encouragement through thick and thin.

To Professor Victor and Mario for their guidance, friendship and open opportunities that have certainly made a great difference in my career so far.

"It is up to the teacher, especially the educator, to remove the obstacles between the person and knowledge, and that he/she feels, in the teacher and educator, how knowledge has transformed that person".

Leandro Karnal

Contents

List of Figures

List of Tables

1 Introduction

This chapter intends to introduce the problem to be addressed in this work. Thus, giving all the main information about the theme and the contextualization, problem, motivation, general objectives, and specific objectives is necessary. In this way, all sections are unspoken to aggregate knowledge about all questions disposed of here.

1.1 Theme Presentation

Time series are well known in the data analysis community since it's a very common type of data structured so that time represents a key factor in the context that the information is inserted (ESLING; AGON, 2012). As time passes, important variations can be seen in data that may highlight meaningful material implicit in raw data, leading to the development of many distinct tasks such as segmentation and clustering. So nowadays, the real applications for time series mining are a reality in many people's lives.

One of the major results derived from time-series analysis is event detection. In this case, a combined change of the variables from successive readings will result in different outcomes for the target variable, which can be seen as a conjuncture by itself that needs particular attention (GURALNIK; SRIVASTAVA, 1999). These events may refer to a known event that happens in real life. Its identification is important for applications or even unknown events classified as outliers and refers to a huge change in data patterns.

The range of possible approaches for event detection is numerous since, according to the scope of the problem, it can drive the researcher to use different techniques. Classification and clustering are very useful when it comes to event detection that, despite some variations in agreement with the problem, are the center of recognition of known events and unknown events, respectively, as seen in (ALIGHOLIAN et al., 2021) and (TEMKO et al., 2009).

Beyond only detecting these events, it is important to predict them before they happen through pattern recognition of how values shift during previously known events.

This is an important topic that can be applied to numerous fields, such as: Predicting earthquakes (MOUSTRA; AVRAAMIDES; CHRISTODOULOU, 2011), substantial increases or decreases in stock market shares (IDREES; ALAM; AGARWAL, 2019), and weather (RADZUAN; OTHMAN; BAKAR, 2013).

The health area is another field that can get good results from the solutions coming from even forecasting. According to recent data observations, these predictions can lead to better options by experts, and actions can be made before worsening the clinical picture or even alerting professionals or users to a possible outcome. Therefore, applying these concepts to problems in this field is another area to be continuously explored since its gains are enormous.

1.2 Contextualization

The concept of event prediction in health matters is already a subject of study well developed in some cases, for example, in predicting strokes in patients (KASABOV et al., 2014) that demonstrate significant improvement when compared to other methods. Also, with the advance and dissemination of IoT (Internet of Things) and ubiquitous computing, sensors became popular among general users providing a big amount of data that can be analyzed to provide insights about the user's health during their routine (SADEGHI; MCDONALD; SASANGOHAR, 2022). Furthermore, Ambient Assisted Living (AAL) aims to provide an ecosystem of different types of sensors, computers, mobile devices, wireless networks, and software applications for personal healthcare monitoring and telehealth systems (MARQUES, 2019).

AAL provides information regarding the subject user and his data according to the characteristics of the objective. Hence, understanding this information and monitoring users' daily activities in AAL can be beneficial for detecting specific events and predicting them before they happen, especially for older people. Moreover, this age group can get real gains from AAL technologies since some events can determine serious consequences for people in this stage of life. For example, preventing strokes, aggravation of symptoms, and worsening of clinical status are some examples of AAL, and time-series event detection and prediction can help. Thus, another aspect that can be analyzed considering the gains for older people is predicting their falling. Falling for people in advanced stages of life may imply serious damage to health for these individuals, having the result aggravated by age (DEGOEDE; ASHTON-MILLER; SCHULTZ, 2003). Furthermore, biomechanical issues and impact on mobility are problems that may derive from falls which can be prevented by using technology to detect and send previous alerts before they happen.

1.3 Problem

The time factor is crucial when dealing with predictions since it's necessary to predict the event in advance to perform actions to prepare for the events before they happen. In addition, the time left between the event prediction and the event occurrences has a strong negative correlation; that is, they go in opposite directions. Therefore, the goal is to predict those events with the necessary time to respond previously, minimizing the number of false alarms along with high-level precision (MEHRMOLAEI; KEYVANPOURR, 2015).

However, when it comes to health matters, the maximization of the time window to act to prevent more serious events ceases to be desirable and becomes crucial. Cases of a false positive or false negative in this scope will result not only in huge effort spent by professionals when no help is needed but in severe consequences since no action may be done during the event. Another important fact that we must mention is related to the outcomes of cases when the event happens but no prediction was made. In this scenario, the user, considering his age, might not be able to reach phones or other means to contact professionals, and the circumstance will become critical. Therefore, providing an accurate prediction and the maximum time window is the key to obtain good results, with caution in dealing with health-related data, while sending these alerts to a competent authority regarding the user's health.

1.4 Motivation

The motivation for this work is to propose an architecture to detect fall events in the home of elderly people based on previous research made by (KALUŽA et al., 2010). Their proposal presents an architecture based on six requirements: monitoring a person and detecting an emergency; being hardware-independent, meaning that it can be easily integrated; the data presentation must be increasingly abstract, allowing an inference about the person in the environment at several abstraction levels; finally, the system must allow the introduction of redundancy by combining several methods, sensors, and viewpoints to solve a given task. In summary, the system must provide an insight into the person and the environment accessible to an observer.

Using the experimental data obtained through the author's research that has already been classified, we are equipped with the data collected and the author model's event detection. In this way, many mishaps are covered, and prediction recognition and pattern detection can be done, leading to different problems to solve than those faced by the original authors. These problems refer to time series prediction, modeling, and interpretation with a whole range of algorithms that can be experimented with. Since the data is already validated by research and scientific publications, this future effort relies on safe information to provide a good technology that experts can use for people in need that fits the scope mentioned.

1.5 General and Specific Objectives

Considering all the previously mentioned arguments regarding the theme presentation, contextualization, problem, and motivation, it's important to highlight all main objectives addressed during this research. Thus, as general objectives, applying time series analysis techniques for event prediction to detect the best algorithms, main parameters, and most suitable data pre-processing treatments and several other questions are deeply studied. Besides, learning more about the state of art for this type of application and comparing results obtained with similar studies are of great value to enrich the scientific community with more cases of study and more conclusions derived from a concise work.

Apart from these points, comprehending how to help older adults from falling is another crucial point of interest, given the coverage of the data set and the specific objective. Therefore, understanding the best time interval for a notification before the event happens while maintaining good accuracy converges for both purposes since the general objectives for this research are going to be applied to a more specific scenario, which is the prevention of older adults from falling. In addition, with the help of a health specialist, identifying the limits and horizons without crossing the ethical line between computing and healthcare is another major point for the application since their knowledge is essential not only for these purposes but also to aggregate more context for the results and the data interpretation.

1.6 Methodology

To reach these goals, a search for open data sets regarding health was explored to identify previous works that conducted studies to collect and analyze information. This investigation has brought a past paper (KALUŽA et al., 2010) to collect data from older adults through a multi-agent system that, in the end, promoted event detection, identifying, for example, when those individuals fell in their homes and the system calls for medical assistance. This way, a great opportunity to go forward with this study was recognized, aiming to manage the same amount of data collected.

Posteriorly, a proposal to predict the events detected and already labeled was initiated through the mapping of academic pieces which approach similar subjects seen in this paper. In addition, earlier research dealing with health information helps us to draw a parallel between strategies for cleaning data and processing the information correctly. Besides, approaches with different algorithms will be used to detect the best parameters and tools to forecast these events with the respective results observed through a perspective of what they mean to the result that a user would have.

This work is organized as follows: In chapter 1 it is presented an introduction to the theme as well as contextualization, the problem, motivation, general and specific objectives and methodology. In chapter 2 we present a literature review of the main points regarding this work. Chapter 3 discusses deeply the works that, somehow, are related to this one and helped in identifying some guidelines to follow. Moreover, in chapter 4 the proposed work and architecture are discussed. In chapter 5 the experimental results and the interpretation of the results are mentioned. Finally, in chapter 6 the conclusion and future works are conferred.

2 Literature Review

This chapter presents all the basic concepts surrounding the research and its impacts on this paper. Thus, the state-of-the-art of all themes that make up the core of this study is explored along with dissertations that guide the paths to prospect. Moreover, it is important to highlight and define scenarios, models, and previous solutions to better understand the advances and different approaches. Finally, section 2.1 briefly introduces notions about Ambient Assisted Living. Following, section 2.2 presents both conceptualizations of the event detection and event forecasting problem. Section 2.3 displays scientific technologies for event forecasting in time series.

2.1 Ambient Assisted Living (AAL)

Assisted living technologies based on ambient intelligence are called ambient-assisted living (AAL) tools. AAL can be used to enhance the wellness of people as well as improve the sense of well-being in routine. Characteristics of its operation may vary depending on the main focus of the proposal. In this way, AAL for medication management tools and medication reminders can help individuals control their health conditions (KHAN et al., 2010), (QUDAH; LEIJDEKKERS; GAY, 2010). This same technology can also act to provide more safety in the environment by monitoring diverse signals of people, devices, or surrounding data in order to analyze emergency events, daily activities, mobility, and automation (DUBOWSKY et al., 2000), (EKLUND et al., 2005), (POLLACK et al., 2003).

Ambient Assisted Living is commonly used to help older people since this technology can operate as an automatic caregiver in some cases for some dependencies. As mentioned in (RASHIDI; MIHAILIDIS, 2012), his survey detects some of the most frequently encountered applications for AAL, for example, Cognitive Orthotics, Continuous Health Monitoring, Therapy, Emergency Detection, Emotional Well-being, and Persuasive Applications. Their ways of addressing the theme may vary, but, for instance, some cases approach each functional form: Wandering prevention tools, Vital Signs Monitoring, Tele-Rehabilitation Systems, Fall Detection, Social Connectedness, and Medical Compliance. Accordingly, it's explicit that there are numerous scientifically proven ways AAL can help older people and its benefits for this public to attend to their needs.

To be specific regarding the type of AAL application studied deeper in this paper, (KALUŽA et al., 2010) propose a multi-agent system composed of multiple interacting intelligent agents. It's worth mentioning that this multi-agent system has various layers. Since ambient intelligence is important for the application, AAL composes a part of the solution implemented by the author. Their strategy aims to prolong the independence of elderly people in reacting to critical situations in case of unusual behavior is detected. Hence, an Emergency Detection approach is addressed in the author's paper. An architecture is presented in Figure 2.1 which reveals the communications between agents to understand better how all agents are connected.

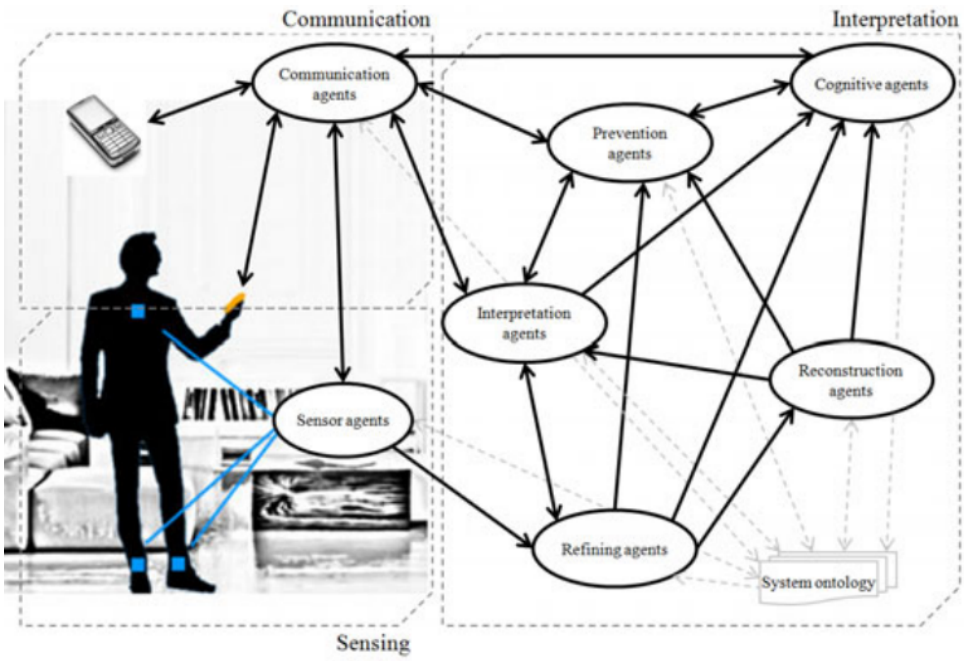

Figure 2.1: Multi-agent system architecture for detecting fell of elderly people. Figure collected from (KALUŽA et al., 2010).

2.2 Event Detection and Forecasting

Event detection is a common problem in the machine learning field, where the problem involves recognizing the change of parameters in the model or even the change of the model itself at an unknown time(s). The problem is widely known as the *change-point detection* problem in the field of statistics (GURALNIK; SRIVASTAVA, 1999). Event detection may vary depending on the number of events that are trying to be detected, for example: Detecting an earthquake or not, which led to two possible options of outcomes for the event, that is, true or false (HUANG et al., 2018). Still, the number of events to be detected can be more than one and, thus, the model needs to be prepared, over time, to detect in which range set the parameters fit in, for instance: Which activity a person is doing at their home (AMARAL; DANTAS, 2017).

Furthermore, event detection can become even more complex by adding another variable to the problem: the uncertainty around the target event to be detected. Depending on the application, different events may not have a specific classification of which points an anomaly occurs. As a result, previous information about the event's labels is unavailable. Consequently, anomalies must be detected through the application's variation of patterns. Problems like these are named anomaly detection (CHANDOLA; BANERJEE; KUMAR, 2009).

Regarding health-related data and event detection, it's important to emphasize that past efforts have been made and were very relevant for the development of this study. In (SILVA et al., 2022), anomalies in heart rate patterns indicate events that may be seen as peaks of stress or anxiety. Hence, detecting these peaks, considering the context and previous patterns of the user, is important for analysis observing these scenarios. Moreover, information regarding the user's location was used to enrich the data and obtain better results through the studies.

Besides event detection problems, event prediction is a step beyond detecting moments. With scenarios already well recognized and validated, a step further in event identification is predicting when already known anomalies or contexts will happen. Ergo, it's important to notice that past knowledge is required for this analysis. Certain variations must be clearly recognized as events, so the forecasting model doesn't get biased.

Examples of event forecasting can be seen in predictions of earthquakes through signals that, as a result, indicate the time remaining for an earthquake with considerable magnitude (ROUET-LEDUC et al., 2017).

To illustrate the flow of an event forecasting problem, Figure 2.2 shows the steps within this type of problem applied to predict weather events. In it, the historical data is collected along with its event data which can be perceived as labels for the events. Thus, the data is then analyzed through a series of techniques such as data cleaning and normalization/scaling, for example, to finally start the experiments with models and determine the best approaches and parameters for the case study.

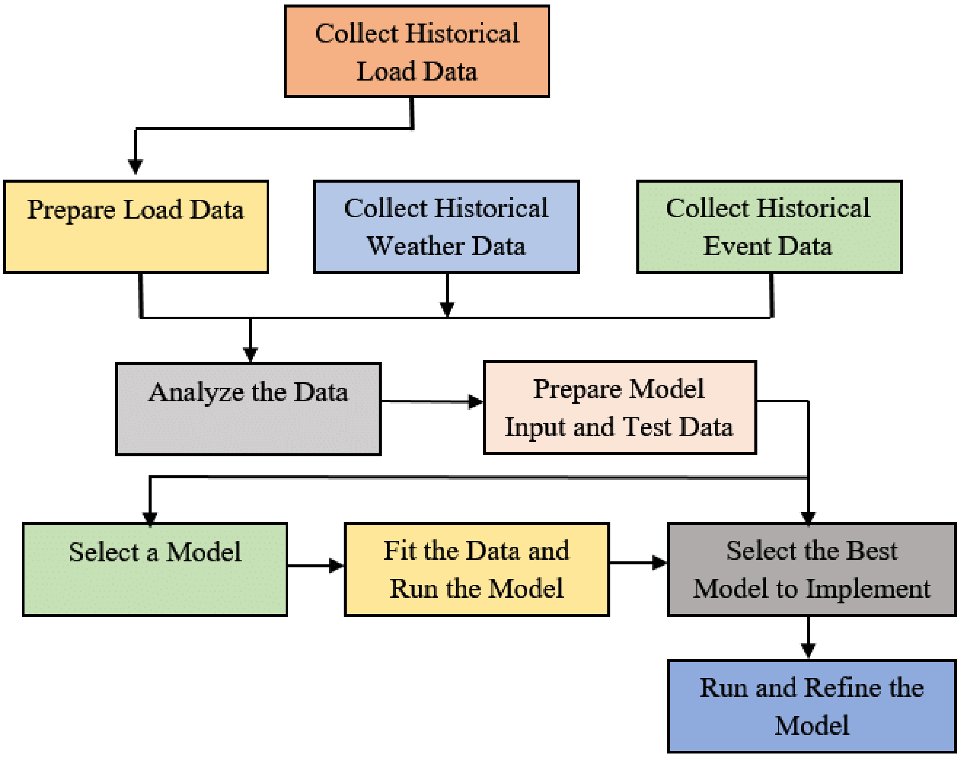

Figure 2.2: Flowchart of a weather event forecasting problem. Figure collected from (MAMUN et al., 2020)

Event forecasting techniques may vary according to the type of data dealt. Geolocation data, images and time series, for instance, all have peculiarities that must be considered in favor of the work's cohesion and results. Therefore, this study focus on predictions regarding time series with its specific technologies for this purpose keeping in mind the health-related data obtained.

2.3 Time Series Approaches for Even Prediction

A time series is a set of measures collected at even intervals of time and ordered chronologically and observed over time (CHATFIELD, 2003). With this in mind, it is easy to perceive that many problems in numerous spheres have time as a crucial variable for their study. Accordingly, forecasting involving time series is very valuable and its study is well disseminated by many academic fields. However, not every time series can be shaped in the previous formal definition due to some reasons:

- **Missing Data**
- **Outlying Data**
- **Unevenly Spaced Time Series**

The first one isn't only a problem for time series but for every data set. Nevertheless, time series missing information, depending on the amount of data lost, can signify a rupture in time flow which may affect deeply the studies. Dealing with this is extensively debated since there are a lot of strategies based, for example, on imputing information and omitting the entire record. These are the most commonly used (BUUREN, 2018). The second one appears very frequently in time series. Methods based on robust statistics can be used to remove these values or incorporate them into the model (MARONNA et al., 2019). The third one happens when data is collected at an irregular time period and, when this occurs, either can be called as unevenly spaced time series or, if big enough, data streams (GAMA, 2010).

Furthermore, time series are basically composed of three aspects: Trends, seasonality and residuals. The trend is the movement that the time series present without considering any irregularities. Seasonality marks specific variations which happen in regular time intervals. Residuals are those values that do not match with the pattern of the data collected, are also known as *outliers* (SHUMWAY; STOFFER; STOFFER, 2000).

Time series have a meaningful irregular component and are not stationary, this is, mean and variance are not constant over time. Therefore, this component turns into one of the most challenging aspects to the model. Because of this, is very difficult to make accurate predictions while many forecasting models try to decompose the target time series into the three mentioned components and predict them separately.

With the development of tools to deal with time series forecasting, the range of possible methods for a prediction in this type of data has been notably growing. Box-Jenkins methodology (BOX et al., 2015) and machine learning regression (MARTÍNEZ-ÁLVAREZ et al., 2015) methods are examples of techniques widely explored, while deep learning approaches, when compared to the other two mentioned, are a more recent mean which has been its understanding exponentially rising due to advances in technology as well as a huge interest by the scientific community to the results generated by its use.

Another key matter of time series forecasting refers to its length. It is well-known that the Bow-Jenkins' models do not work well for long time series mainly due to the time-consuming process of parameters optimization and to the inclusion of information which is no longer useful to model the current samples (MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 2008). Although a preliminary approach to use a distributed ARIMA model has been recently published, it remains challenging to deal with such time series with classical forecasting methods. However, a number of machine learning algorithms adapted to deal with long time series have been published in recent years (WANG et al., 2022). Ergo, dealing with these problems not only depends on the structural components of the time series but also needs to be considered its length to find the best possible approach.

3 Related Works

This chapter explores works that contributed to enhancing the research done here using results obtained, techniques, and tools used along with the chosen approaches. Moreover, the works are briefly described, detailing their achievements and negative and positive aspects, along with a concise description of how that scientific piece contributed to the work presented here. At the end of this section, a table is presented to compare the papers mentioned and their main points while showing which key points are exploited as an advance based on these outcomes.

3.1 AAL activity recognition

One of the main points of detecting events based on activity recognition refers to the data collected and the accuracy of the prediction model results. Since this type of proposal deals with information generated by individuals through sensors during their routine, imprecision and irregularities might be found, negatively impacting the results. In (PATEL; SHAH, 2019), considering these key points, the paper aims to develop a guide to select the best approaches to identify human activity in an AAL environment.

The methodology used to conduct this study consisted of many phases. In the first phase, the materials required for the research are collected using electronic and manual library searches. The keywords employed for the search were: Activities of daily living, anomaly detection, smart home, elderly care, assisted living, and ambient intelligence for AAL. The authors used different databases for academic papers, such as DBLP computer science bibliography, Compendex, Mendeley, IEEE, Springer, and Elsevier from various universities. The second phase consists of making a list of references, citation count, and ranking for inclusion or exclusion of each research article. Then, the researchers identify seventeen types of challenges of activity recognition to investigate the available literature. The categorization of the groups of challenges for activity recognition, along with some characteristics of the technologies used, was very helpful in perceiving possible gaps in the approaches used in many articles. Also, obtain a critical view of the data used in this research, obtained from a previous effort made by another researcher.

The conclusion of the work described points out how plentiful solutions made so far still have three main objectives to study: Recognize multiple residents, consider individuals at risk, and change themselves because of users' intrinsic requirements.

3.2 Learning from imbalanced data: Open challenges and future directions

Since this work aims to deal with the problem of detecting previously the falling events that might happen to elderly people in their homes, it is expected to have more events without falling rather than with falling. In this way, understanding the discussions around the theme along with the gathering of information is essential. That said, (KRAWCZYK, 2016) presents a survey addressing some open challenges and future directions regarding imbalanced data and its intrinsic problems.

Seven vital areas of research were identified during the analysis: Classification, Regression, Clustering, Data Streams, Big Data Analytics and Applications and Computer Vision. Considering binary classification, it is surely one of the most explored branches, yet, some topics still are difficulties encountered when managing this type of data structure. Therefore, the following future directions of research were disclosed:

- **Analyzing the structure of classes**: Studying the neighborhood of each minority class and assigning to a group allows a new perspective about the problem. Moreover, classifiers that can incorporate the background knowledge about objects into their training procedure and preprocessing ideas that select important or difficult samples to concentrate are relevant subjects to consider as adversities.

- **Extreme class imabalance**: Methods like SMOTE can deteriorate the classification performance. Using randomized methods may not be advisable too since it has a high potential variance induced by the imbalanced ratio. Therefore, techniques that empower the minority class and predict or reconstruct a potential class structure are a promising direction.

- **Classifier's output adjustment**: Methods that will be able to take into consideration the characteristics of classified examples and adjust the classifier‘s output individually for each new object. Furthermore, output adjustment is considered as an independent approach. Yet from a general point of view it may be fruitful to modify outputs even when data-level or algorithm-level solutions have been previously utilized. This way we may achieve class balancing on different levels, creating more refined classifiers. Analyzing the output compensation may also bring new insight into supervising undersampling or oversampling to find balanced performance on both classes.

- **Ensemble Learning**: There are no clear indicators on how large ensembles should be constructed. Usually, their size is selected arbitrarily, which may result in some similar classifiers being stored in the pool. It would be beneficial to analyze the relations between the characteristics of an imbalanced dataset and the number of classifiers required to efficiently handle it.

This work helped to comprehend the dimension of some problems that, despite might seem ordinary and well explored, still needs attention and careful decisions to consider. Thus, during implementation, these information raised important questions in decision-making.

3.3 Ensemble learning for software fault prediction problem with imbalanced data

Imbalanced classes have significant effects on the results that a classifier might display as output due to the predominance of the majority class. Hence, (KHUAT; LE, 2019) aims to integrate the sampling techniques and common classification techniques to form a useful ensemble model for predicting software defects.

Therefore, the imbalanced data set goes through a sampling part to divide the data for numerous branches where they get there already balanced. Then each branch trains a classifier that works as the real classifier for new data incoming, as an ensemble classifier. During the training process, the majority class samples in the original imbalanced dataset are split into several bins by adopting the random undersampling method. Each bin includes the equal number of patterns to that of the minority class, and then all minority class patterns are put into each bin to form the balanced training dataset. After that, each base classifier will be trained on a separated balanced dataset by a specific classification algorithm. Finally, the final classifier is built by combining the outcomes of base predictors relied on the majority voting rule.

The results obtained by the authors show that the combination of the sampling technique and ensemble learning contributes to forming a promising classifier for the software fault prediction problem. Moreover, the steps taken during the experimental results showed also a diverse range of possible analyses to follow along with metrics to contemplate. Furthermore, it is important to mention that this work also deals with predicting events with imbalanced data and is a good parameter to follow.

3.4 SMOTE Variants for Imbalanced Binary Classification: Heart Disease Prediction

Strategies to overcome the obstacles displayed by the few occurrences of positive or negative classes in binary classification are one of the first and main concerns of researchers. Ergo, techniques for oversampling, undersampling or combined under and oversampling might result in a good solution. In (ZHENG, 2020), different variants of SMOTE (*Synthetic Minority Oversampling Technique*) are implemented and tested in data regarding heart disease in order to predict these cases.

Then, Borderline-SMOTE, SVM-SMOTE, KMeans-SMOTE are used in experiments to present the best technique for this purpose. It is crucial to highlight that since the authors are dealing with health information, a similar direction can be applied to our ambitions in this paper. During their experiments, SVM-SMOTE and Borderline-SMOTE outperform other SMOTE variants for the type I error, this is, the number of false positives.

Thus, this work presented a good indication of which technique to implement in order to resolve some problems arising of types of error. In this work, despite the intention being different, reducing the number of false negatives while increasing the number of true positives, a consolidated background of these techniques was collected.

3.5 Addressing binary classification over class imbalanced clinical datasets

Understanding the performance of some classifiers to deal with imbalanced datasets as well as their influences while varying balancing techniques is another key point to perceive. Therefore, in (KUMAR et al., 2022) carries the efficacy of six different algorithms for binary classification over imbalanced data: Decision Tree, K-Nearst Neighbor, Logistic Regression, Artificial Neural Network, Support Vector Machine and Gaussian Naïve Bayes over distinct information regarding health data. Breast Cancer Disease, Coronary Heart Disease, Indian Liver Patient, Pima Indians Diabetes and Coronary Kidney Disease are the various health problems addressed in this work. Simultaneously, seven different tools for balancing information are tested: Undersampling, Random oversampling, SMOTE, ADASYN, SVM-SMOTE, SMOTEEN and SMOTETOMEK.

The wide range of algorithms, techniques for balancing information and datasets gathered by this work were very useful during the construction of this study since it promoted a discovery concerning techniques to be used during the experiments of this paper. Hence, the results founded that SMOTENN balacing method ofter performed better over all the other six data-balancing techniques with all six classifiers and for all five clinical datasets. Furthermore, all techniques achieve similar results during the experiments except for SMOTEENN.

3.6 Comparative Table

With the works presented here, it is imperative to highlight the gaps that this work aims to clarify and get deppen. Therefore in Table 3.6 presents a comparative table between the works and some of the themes addressed by them, contrasting with this work that approaches, somehow, all of the points shown.

	AAL	**Balancing Techniques**	**Binary Classifiers**	**Health Information**	**Event Prediction**
Work 1	X			X	
Work 2		X	X		
Work 3		X	X		X
Work 4		X	X		
Work 5		X	X	X	X

4 Proposed Work

The main goal of this chapter refers to explain and clarify all the aspects of this work that are important for understanding the technologies and the study developed here. Therefore, the chapter is divided so that the process overview is presented, followed by describing each module that composes the architecture design. Therefore, the intention is to evaluate different types of algorithms with the target of predicting elderly falls with good precision and enough time to act before the event happens.

In Figure 4.1, it can be seen an image that represents the process studied and its main layers. Firstly, the data was collected through a study conducted in (KALUŽA et al., 2010) where some participants, in a test environment, wearing sensors based on a multi-agent architecture got the data referring to the position of each sensor obtained, then labeled the information of each moment in the time-series with an event that, among others, there is the falling event. All this content represents a work previously consolidated by (KALUŽA et al., 2010), and the labeled data for every moment are already set.

Moreover, the event prediction layer represents a step forward in Kaluvza's work since it's another technology used to predict certain events. In addition, from the techniques adopted in the previous studies, a guided case study can be done with some algorithms and statistical approaches that might get better results due to the experimentation in consolidated scientific works.

Hence, directing the different results obtained with the main matters of this study, i.e., window time and precision, is the central point to achieve. Which algorithm got, from a perspective of "Time to react" X "Accuracy" has the best outcome.

It is important to highlight that some changes were made regarding the label of the fall event. In order to predict the events the label is shifted according to the time interval between the event and the prediction desired. Thus, all events are grouped second by second and the prediction varies. For example, for a prediction of the event that is going to happen 5 seconds after, the labels of each second are shifted 5 times. Then, in one line we are going to have the data from that specific second but the target variable, this is, the label, from 5 seconds after. Therefore, the model is tested for events in the future instead of the current event.

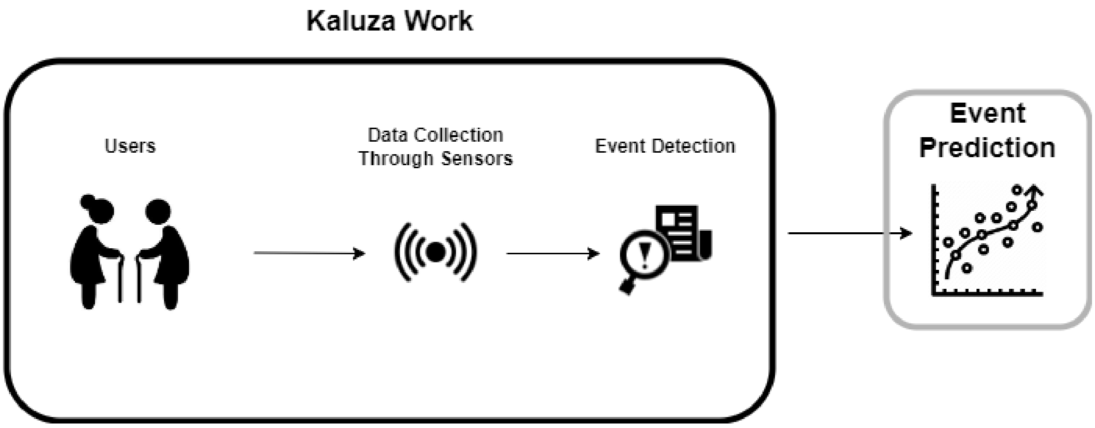

Figure 4.1: Proposed Work Overview.

4.1 Event Prediction Architecture

To better comprehend the event prediction projected in this work, it is important to clarify its core's main aspects. Thus, Beyond just adding a new compartment, this research designs an architecture for predicting the falling events in elderly people's homes based on the Fog-Cloud paradigm. But, it is necessary to point out the reasons for such an option since a few other distributed systems paradigms can be used.

A Fog-Cloud architecture is properly robust to deal with the amount of data generated by the sensors used to measure users' activities since there is a reduction in the possibility of network bottlenecks with the processing closer to the data extraction while simultaneously enhancing the responsiveness of the system (DENG et al., 2016). The data generated by the sensors are produced by the multi-agent architecture proposed by (KALUŽA et al., 2010), which, in this case, is considered the Edge layer of the architecture. Moreover, concerning the model life-cycle used for the event prediction, concepts of the Lambda architecture are also used with the information being sent to two parallel workflows. As mentioned in (KIRAN et al., 2015), the lambda architecture is centered on providing high scalability and having a low delay in both data queries and updates, and also able to deal with either human or hardware errors. To illustrate the flux and how the information flows through each part, Figure 4.2 presents an overview of the process.

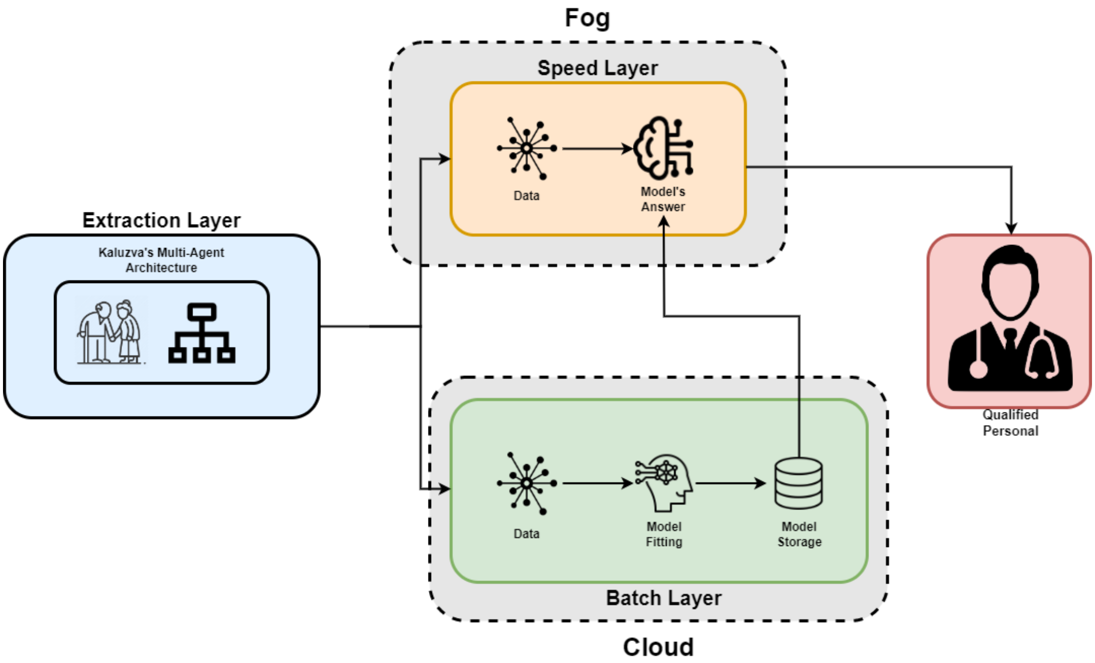

Figure 4.2: Proposed Architecture

First, the Extraction Layer is responsible for promoting the extraction from Kaluzva's multi-agent architecture. Since the data has a processing part within Kaluzva's architecture, there's no need for deep data cleaning; thus, it can be moved on to the next phases. In this part, it is essential to highlight that the information flows in two different ways, in parallel and simultaneously. The Batch Layer, which is contained in the Cloud, is responsible for processing the information and using it to fit a new model for the future since, at a certain moment, the models will be outdated with new information arriving along with new contexts. Hence, these recent models are stored in a proper database for future usage when the amount of fresh data is enough to delimit a new context. Another aspect that is crucial to mention is the fact that this whole process is enclosed in the Cloud because it requires a more robust storage power and doesn't need speed. For these reasons, it can be physically located far from the final user.

The other component, the Speed Layer, is responsible for performing a quick and fast answer for the events that are happening at the current moment from the information gathered previously. This way, the data is passed as input for the model to obtain the prediction. Hence, it is easy to perceive that this layer must be agile to provide the predictions as soon as possible. For this purpose, this layer is in the Fog node because of its need for speediness, and since the Fog is closer to the final user, there is less latency time. Another important matter is the possibility of choosing the prediction time-window, such as a prediction to check if a fall will happen in 2 minutes instead of 1 minute. This type of setting is possible thanks to the Fog's responsiveness in reducing the time needed to perform such different configurations.

At last, the prediction's result is sent to a specific and qualified health authority to choose the best course plan for the situation according to the outcome generated by the technology and his proper knowledge. It is crucial to remark that this new technological approach aims to be a new tool to help health specialists in the care of their elderly patients and not a form of final answer regarding users' conditions.

As an outcome of the proposed architecture, a notification alerting about a possible falling event will not avoid it from happening, but it will surely provide a wider time window of action. A similar architecture was proposed in a previous work (SILVA et al., 2022). Therefore, the solution promoted here can be seen as feasible and promising because of past investigations, but this time, it is applied to a different ending, which is the event prediction.

5 Experimental Results

This chapter describes all the facets of the dataset, its processing, features, and the results obtained. The data was collected in (KALUŽA et al., 2010), and it refers to the experimentation to evaluate the multi-agent proposal for detecting falling events in elderly people's homes.

The data comprises 164,860 measures of events with a granularity of milliseconds collected via experiments made with five different people during five sections each, thus, 25 datasets. The data has information about the candidate and the section; for example, A01 refers to candidate A and his first experiment. Also, each measurement has a label defining which sensor was responsible for collecting that information *(left ankle, right ankle, belt, or chest)* and its values for the X, Y, and Z axis in the specified timestamp. Moreover, the activity recognized by the multi-agent architecture for that sensor is also provided and labeled as *walking, sitting down, sitting, standing up from sitting, falling, lying, standing up from lying, lying down, sitting on the ground, standing up from sitting on the ground or all fours.*

Another worth mentioning is that each experiment has a different number of measures and, therefore, a different number of occurrences of falling. Table 5 shows each candidate's distribution of measures and falling events in each experiment. The first number refers to the number of moments registered, and the second refers to the number of fall events.

	Candidate A	**Candidate B**	**Candidate C**	**Candidate D**	**Candidate E**
Experiment 1	**5830 / 80**	**6647 / 127**	**6297 / 82**	**7463 / 131**	**9098 / 100**
Experiment 2	**5691 / 96**	**6091 / 101**	**6216 / 72**	**6416 / 137**	**8075 / 198**
Experiment 3	**5326 / 94**	**5670 / 95**	**6267 / 81**	**6562 / 131**	**9112 / 215**
Experiment 4	**5402 / 93**	**5789 / 84**	**6323 / 123**	**5454 / 121**	**8514 / 165**
Experiment 5	**5224 / 83**	**5953 / 88**	**6490 / 115**	**6057 / 111**	**8893 / 247**

Table 5.1: Table with the number of measures / falling events for each candidate and experiment

Ergo, the dataset composition is described along with its features, observed through a Knowledge Discovery Database method. The tests conducted by Kaluzva were made during 5 minutes of experiments maximum and 3 minutes minimum, varying from candidate and experiment. Furthermore, the continuity of the work was also investigated by locating the first and last time stamp from each experiment and each candidate. Also, each experiment had a little interruption from experiment to experiment for every candidate.

5.1 Data Preprocessing

Since the proposal was designed to be singular and adaptable for the user, the dataset was divided according to the candidate, joining each experiment separately. Therefore, five datasets, A, B, C, D, and E, were made from the original information. Figure 5.1 shows how the data came originally.

sequence_name	date_format	x	y	z	belt	chest	left_ankle	right_ankle	activity
A01	2009-05-27 14:03:47.933	3.156413	1.658062	0.031690	0	0	1	0	falling
A01	2009-05-27 14:03:47.960	2.978983	1.738826	0.371500	0	0	0	1	falling
A01	2009-05-27 14:03:47.987	3.203583	1.513173	1.083314	0	1	0	0	falling
A01	2009-05-27 14:03:48.013	3.298423	1.501675	0.721679	1	0	0	0	falling
A01	2009-05-27 14:03:48.040	3.164907	1.672547	0.092952	0	0	1	0	falling
A01	2009-05-27 14:03:48.093	3.258224	1.424610	1.176932	0	1	0	0	falling
A01	2009-05-27 14:03:48.120	3.547822	1.175537	0.401623	1	0	0	0	falling
A01	2009-05-27 14:03:48.150	3.175236	1.694705	0.067486	0	0	1	0	falling
A01	2009-05-27 14:03:48.177	3.014699	1.681337	0.201258	0	0	0	1	falling
A01	2009-05-27 14:03:48.203	3.419891	1.463586	1.195048	0	1	0	0	falling
A01	2009-05-27 14:03:48.257	3.184952	1.699849	0.168956	0	0	1	0	falling
A01	2009-05-27 14:03:48.310	3.666966	1.400693	1.070500	0	1	0	0	falling

Figure 5.1: Dataset Example

However, to validate the proposed architecture, a different time stamp had to be implemented to have more similar information to what is expected to collect. This way, a new granularity was introduced based on the type of test performed, for example, second by second. Furthermore, in order to make tests and experiments, the data were aggregated second by second, and the interval aimed to predict varies according to the test.

To verify if this wouldn't affect the time intervals mentioned during the end of an experiment and the beginning of a new one for the same candidate, the activity performed by them was analyzed since this new timestamp with a new granularity could cause an interruption in the activity. For example, suppose at the end of the first experiment for candidate A, the person was falling, and at the beginning of the second experiment for the same candidate, the person was walking. In that case, this wouldn't be acceptable in terms of logical sense and might bias the study. Therefore, we analyzed the dataset and observed that every end of each experiment and the beginning of a new one were initialized with the perceived walking activity. Hence, we do not inflict disruption in the logic of the activity performed by adopting a new timestamp.

Furthermore, a special treatment had to be implemented regarding the information from sensors in the X, Y, and Z axis. Figure 5.2 shows that a different range of variation can be perceived according to the sensor. Therefore, the same measure for the X axis, but collected from different sensors, might have another mathematical significance. To overcome this, an approach of scaling the information was performed in the dataset in agreement with the candidate and the sensor. It is important to mention that the scaling process used was the *Robust Scaler*. The dataset contains numerous outliers which need to be considered without causing an impact on the scaling process of the other measurements, so the significance of each measure is preserved, and the data is confined to a new range to help the estimators used.

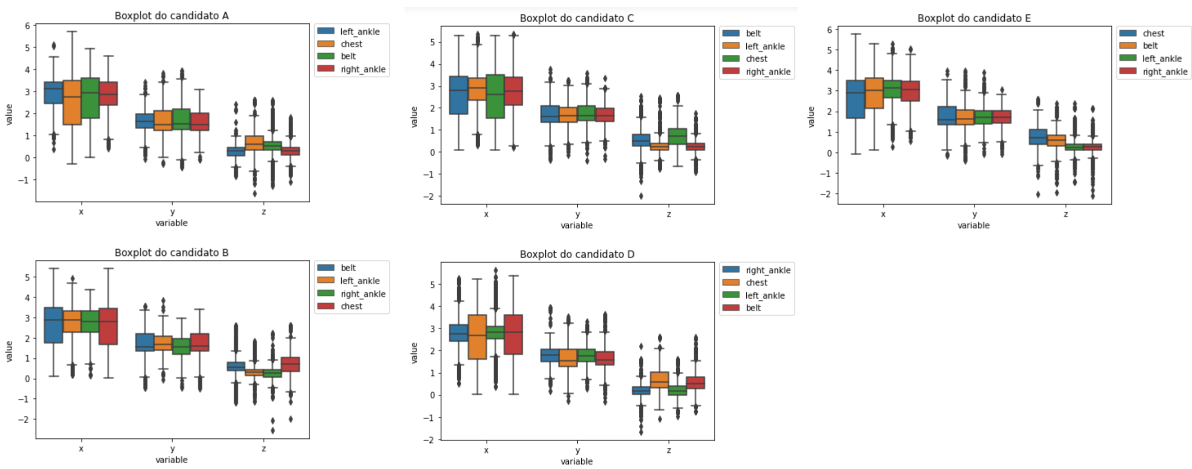

Figure 5.2: Sensor Variation for Each Candidate

Moreover, selecting a measurement to represent the context at the given moment is necessary to transform the timestamp. So, as a project decision, we decided to group the information according to the candidate and the time interval. If a falling event happens at that given time interval, it is considered that the context at that moment was of *falling*.

For this reason, each group median, standard deviation, mean, minimum, and maximum are settled in the dataset to compose the dataset used in the experimentation. Figure 5.3 shows a sample of the final dataset for a specific candidate after all the transformations and preprocessing mechanisms.

	x					y					z					current_falling	future_falling
	mean	median	std	min	max	mean	median	std	min	max	mean	median	std	min	max		
2022-10-03 12:37:40	0.934212	0.829064	0.251266	0.664626	1.522403	0.467773	0.439092	0.173210	0.115012	0.755224	0.741662	0.728618	0.540709	-0.691079	1.319416	0	0.0
2022-10-03 12:37:41	0.905135	0.780450	0.278802	0.681674	1.621213	0.532310	0.502916	0.182067	0.127702	0.812372	0.819816	1.075950	0.560124	-0.304198	1.529938	0	0.0
2022-10-03 12:37:42	0.840742	0.772393	0.232777	0.624243	1.627052	0.387253	0.392164	0.183630	-0.035553	0.823203	0.655113	0.928247	0.568974	-0.444601	1.323195	0	0.0
2022-10-03 12:37:43	0.326826	0.313951	0.559402	-1.362884	0.959405	0.667365	0.480440	0.495692	0.071199	2.137215	0.622779	0.666697	0.966075	-1.442677	3.122116	0	0.0
2022-10-03 12:37:44	0.147184	0.171348	0.167381	-0.139681	0.627150	0.465143	0.471775	0.221539	-0.214249	0.881811	-0.132059	0.363678	0.938349	-1.727601	0.888822	0	0.0
2022-10-03 12:37:45	0.133467	0.088619	0.171624	-0.012387	0.936462	0.617891	0.675842	0.370342	-0.952796	1.372807	-0.007805	0.304691	1.027937	-1.517675	4.000016	0	0.0
2022-10-03 12:37:46	0.099029	0.050149	0.107246	-0.002635	0.346362	0.839285	0.783394	0.281393	0.362374	1.325538	-0.095489	0.339548	0.799651	-1.454401	0.850109	0	0.0
2022-10-03 12:37:47	0.113434	0.068296	0.123014	-0.099101	0.447357	0.759021	0.779018	0.424674	-0.093226	1.319642	0.065021	0.486017	0.762119	-1.964810	0.902690	0	0.0
2022-10-03 12:37:48	0.176284	0.084081	0.160310	0.001201	0.552247	0.783785	0.832318	0.407035	0.141963	1.421611	-0.069611	0.389943	0.791396	-1.220643	0.810908	0	0.0
2022-10-03 12:37:49	0.147118	0.143754	0.098916	-0.043152	0.321226	0.759568	0.924769	0.394223	0.147244	1.423988	-0.210122	0.122980	0.911856	-1.489717	0.972170	0	0.0
2022-10-03 12:37:50	0.115893	0.072812	0.096650	-0.018520	0.321434	0.811048	0.930665	0.292277	0.283310	1.223610	-0.005441	0.561954	0.863480	-1.298878	0.939372	0	0.0
2022-10-03 12:37:51	0.111755	0.068355	0.092097	-0.007865	0.311713	0.760113	0.916586	0.306521	0.234465	1.142246	-0.154297	0.463269	0.878603	-1.464362	0.774623	0	0.0

Figure 5.3: Dataset After the Preprocessing Methods

Ergo, the index of the dataset is the new timestamp value settled according to the chosen grouping, in the case of Figure 5.3, second by second. The X, Y, and Z measures were grouped by collecting the group's mean, median, standard deviation, minimum, and maximum. In addition, the *current falling* column was set according to the event, or not, of falling in that time interval, while the *future falling* represents the label of the next event. Thus, if a falling event happened, it is classified as '1' and, if not, '0'.

Another important factor to highlight is the prediction based on the dataset. To perform this, the label is shifted to obtain the *future falling* measure, which represents the next occurrence of the event. Thus, the target variable is the *future falling* which determines whether the model can predict a falling event or not.

5.2 Experiments

This section is going to discuss the experiments conducted and their results. Moreover, all aspects regarding algorithms, hyperparameters, balancing techniques, and design decisions are deeply discussed to clarify every facet of this implementation and its characteristics. Furthermore, it is worth mentioning that a random state parameter was set in all the tests to guarantee its replicability.

5.2.1 First Experiment

In this first experiment, a series of classic algorithms were tested in order to obtain a baseline to guide further experiments. Therefore, Logistic Regression, KNeighbors Classifier, Decision Tree Classifier, and Random Forest Classifier were tested in a pipeline conducting a Robust Scaler along with the classifier itself. The preprocessed information, firstly, is divided into train and testing. Moreover, a Grid Search with cross-validation was performed to study a wide range of hyperparameters that resulted in almost 50 combinations for each classifier trying to predict the events for 5, 10, and 15 seconds before they happen.

The results were not satisfactory enough during this first experiment since no specific technique to deal with imbalanced data was implemented. To evaluate the results, the best estimator encountered was defined as the one with the best results for the F1 metric. However, other metrics were also gathered: ROC AUC score, Recall, Precision, and Balanced Accuracy.

Since Grid Search CV defines the best hyperparameters for the algorithm, a nested cross-validation with the best estimator is performed to get a more realistic result from the best estimator. The results are then displayed as well as the confusion matrix for the tests with the test sample. The results are shown in Figures 5.5, 5.6 and 5.7. They represent the best variation of hyperparameters for each technique performed in the test sample. Moreover, 5.4 shows the results obtained during this experiment for the train data. The first, second and third number in each cell represents the result for the times of 5, 10 and 15 seconds, respectively.

Logistic Regression

- **Random State**: 1007
- **dual**: False
- **class weight**: Balanced
- **Fit intercept**: [True, False]
- **Solver**: Liblinear
- **Penalty**: [l1, l2]
- **Multi class**: ovr
- **max iter**: [100, 200, 300, 400, 500]

K-Neighbors Classifier

- **N Neighbors**: [3, 5, 7, 9, 10, 12, 15]
- **Weights**: [uniform, distance]
- **algorithm**: [ball tree, kd tree, brute]

Decision Tree Classifier

- **Critetion**: [gini, entropy, log loss]
- **Max Depth**: [2, 3, 5, 7]
- **Min Samples Split**: [2, 3, 5, 7]
- **Min Samples Leaf**: [2, 3, 5, 7]
- **Max Features**: auto
- **Random State**: 1007
- **Class Weight**: balanced

Random Forest Classifier

- **Critetion**: [gini, entropy, log loss]
- **Max Depth**: [2, 3, 5]
- **Min Samples Split**: [2, 3, 5, 7]
- **Min Samples Leaf**: [2, 3, 5, 7]
- **Max Features**: auto
- **Random State**: 1007
- **Class Weight**: balanced

<table>
<tr><th colspan="6">Logistic Regression</th></tr>
<tr><th></th><th>AUC</th><th>RECALL</th><th>PRECISION</th><th>F1</th><th>BALANCED ACCURACY</th></tr>
<tr><td>A</td><td>76.6 | 72.7 | 77.4</td><td>90 | 80 | 75</td><td>7.7 | 5.8 | 5.7</td><td>14.1 | 10.8 | 10.6</td><td>78.1 | 69.6 | 68.4</td></tr>
<tr><td>B</td><td>75.6 | 62.9 | 59.5</td><td>76.7 | 68.3 | 58.3</td><td>6.3 | 3.4 | 3.5</td><td>11.5 | 6.4 | 6.6</td><td>74.3 | 60.2 | 58.7</td></tr>
<tr><td>C</td><td>84.7 | 71.4 | 72.3</td><td>76.7 | 68.3 | 73.3</td><td>8.2 | 4.7 | 5</td><td>14.8 | 8.7 | 9.4</td><td>77.5 | 65.5 | 69.7</td></tr>
<tr><td>D</td><td>64.6 | 66.4 | 76</td><td>65 | 70 | 70</td><td>5.2 | 4 | 4.5</td><td>7.7 | 7.7 | 8.4</td><td>66 | 61.6 | 64</td></tr>
<tr><td>E</td><td>81.5 | 68.6 | 60.1</td><td>90 | 61.7 | 50</td><td>5 | 3.1 | 2.4</td><td>9.5 | 5.9 | 4.5</td><td>79.1 | 63.7 | 56</td></tr>
</table>

<table>
<tr><th colspan="6">K Neighbors Classifier</th></tr>
<tr><th></th><th>AUC</th><th>RECALL</th><th>PRECISION</th><th>F1</th><th>BALANCED ACCURACY</th></tr>
<tr><td>A</td><td>60.3 | 54.3 | 61.8</td><td>10 | 0 | 0</td><td>40 | 0 | 0</td><td>16 | 0 | 0</td><td>54.9 | 49.8 | 49.4</td></tr>
<tr><td>B</td><td>53.4 | 58.2 | 54.9</td><td>0 | 0 | 0</td><td>0 | 0 | 0</td><td>0 | 0 | 0</td><td>49.9 | 49.9 | 49.9</td></tr>
<tr><td>C</td><td>69 | 49.6 | 53.3</td><td>6.7 | 0 | 0</td><td>20 | 0 | 0</td><td>10 | 0 | 0</td><td>53.1 | 49.9 | 49.8</td></tr>
<tr><td>D</td><td>60.5 | 70.4 | 63.8</td><td>0 | 10 | 5</td><td>0 | 13.3 | 10</td><td>0 | 10.7 | 6.7</td><td>49.5 | 54.2 | 51.8</td></tr>
<tr><td>E</td><td>57.6 | 50.5 | 49.9</td><td>0 | 0 | 0</td><td>0 | 0 | 0</td><td>0 | 0 | 0</td><td>49.8 | 49.7 | 49.9</td></tr>
</table>

<table>
<tr><th colspan="6">Decision Tree Classifier</th></tr>
<tr><th></th><th>AUC</th><th>RECALL</th><th>PRECISION</th><th>F1</th><th>BALANCED ACCURACY</th></tr>
<tr><td>A</td><td>67.9 | 67 | 59</td><td>85 | 73.3 | 63.3</td><td>5 | 5.9 | 4.5</td><td>9.5 | 10.9 | 8.4</td><td>67.9 | 67.8 | 59.5</td></tr>
<tr><td>B</td><td>58.2 | 58.7 | 51.5</td><td>56.7 | 58.3 | 50</td><td>3.8 | 3.1 | 2.9</td><td>7.1 | 5.8 | 5.5</td><td>59.9 | 56.3 | 53.7</td></tr>
<tr><td>C</td><td>75.7 | 72.4 | 62.5</td><td>88.3 | 56.7 | 56.7</td><td>5.8 | 6.1 | 4.6</td><td>10.9 | 10.7 | 8.4</td><td>76.3 | 65.2 | 61.6</td></tr>
<tr><td>D</td><td>68.1 | 62.9 | 62</td><td>75 | 85 | 65</td><td>5.4 | 3.9 | 4.8</td><td>10 | 7.5 | 8.8</td><td>68.9 | 62.9 | 61</td></tr>
<tr><td>E</td><td>75.3 | 68.8 | 53.8</td><td>80 | 85 | 48.3</td><td>4 | 3.2 | 1.9</td><td>7.7 | 6.2 | 3.59</td><td>73.4 | 69.4 | 51</td></tr>
</table>

<table>
<tr><th colspan="6">Random Forest Classifier</th></tr>
<tr><th></th><th>AUC</th><th>RECALL</th><th>PRECISION</th><th>F1</th><th>BALANCED ACCURACY</th></tr>
<tr><td>A</td><td>83.4 | 78.4 | 78.7</td><td>50 | 38.3 | 48.3</td><td>7.5 | 4.9 | 6.2</td><td>13 | 8.6 | 11</td><td>67.3 | 58.7 | 62.9</td></tr>
<tr><td>B</td><td>84.8 | 68.3 | 68.7</td><td>46.7 | 16.7 | 28.3</td><td>7.6 | 3 | 5</td><td>12.9 | 5 | 8.4</td><td>66.4 | 50.3 | 56.3</td></tr>
<tr><td>C</td><td>86 | 78.2 | 78.8</td><td>50 | 21.7 | 36.7</td><td>14 | 7.1 | 5.9</td><td>21.4 | 10.5 | 10</td><td>71 | 57.5 | 62.6</td></tr>
<tr><td>D</td><td>69.4 | 75.7 | 70.6</td><td>35 | 50 | 35</td><td>9.2 | 6.6 | 5.5</td><td>14.3 | 11.6 | 9.4</td><td>63.1 | 65.3 | 58.9</td></tr>
<tr><td>E</td><td>87.8 | 73 | 69.3</td><td>48.3 | 31.7 | 15</td><td>12.2 | 4.5 | 1.8</td><td>18.2 | 7.8 | 3.2</td><td>71.1 | 60 | 49.5</td></tr>
</table>

Figure 5.4: Results Obtained from the First Experiment

After the results of the first experiment with a time interval of 5 seconds before the event happens the outcomes shown some developments with good potential according to some metrics. The results did not vary significantly between the candidates while the most consistent algorithm for this time interval was the Logistic Regression. However, it is important to pay attention to the Recall, F1 and Balanced Accuracy. First, the

Logistic Regression				
	TP	TN	FP	FN
A	1	202	61	7
B	5	195	98	3
C	7	212	96	0
D	8	220	89	0
E	8	87	340	0

K-Neighbors Classifier				
	TP	TN	FP	FN
A	0	263	0	8
B	0	293	0	8
C	0	308	0	7
D	0	309	0	8
E	0	427	0	8

Decision Tree Classifier				
	TP	TN	FP	FN
A	2	216	47	6
B	8	107	186	0
C	0	288	20	7
D	8	69	240	0
E	8	216	211	0

Random Forest Classifier				
	TP	TN	FP	FN
A	3	191	72	5
B	0	293	0	8
C	0	288	20	7
D	8	69	240	0
E	8	199	228	0

Figure 5.5: Results Predicting 5 Seconds After

Recall metric showed good results and indicates that the algorithms tested, excepts for K Neighbors Classifier, are dealing well in detecting the positive events, that is, fall events. Yet, F1 and Balanced Accuracy are showing two different perspectives: While F1 shows a poor result, Balanced Accuracy points out to a great outcome. This is related to the type pf analysis made by these two metrics.

Recall shows that the number of False Negative results measured in the training data is low while Precision points that the number of False Positive is high. With this in mind, F1 gets a poor result since does not take into consideration the true negative rate, this is, specificity. On the other hand, Balanced Accuracy gets a good result since a considerable number of cases predicted right as True Negatives is made.

Logistic Regression				
	TP	TN	FP	FN
A	6	121	141	2
B	2	168	124	6
C	6	193	113	2
D	2	291	16	6
E	8	0	426	0

K-Neighbors Classifier				
	TP	TN	FP	FN
A	0	262	0	8
B	0	292	0	8
C	0	306	0	8
D	0	307	0	8
E	0	426	0	8

Decision Tree Classifier				
	TP	TN	FP	FN
A	4	138	124	4
B	2	218	74	6
C	8	113	193	0
D	8	0	307	0
E	6	165	261	2

Random Forest Classifier				
	TP	TN	FP	FN
A	7	76	186	1
B	2	208	84	6
C	0	306	0	8
D	8	0	307	0
E	6	165	261	2

Figure 5.6: Results Predicting 10 Seconds After

Moving forward in the experiment, as expected, the results maintain their shape during the predictions for a 10 seconds interval. K Neighbors Classifier got terrible results, even worse than the previous one made with a 5 second interval. The other algorithms, almost all, got a lower performance when compared to the past results. Logistic Regression and Decision Tree Classifier, especially, achieve the best results as it was in the first time interval and Random Forest Classifier got a slight improvement regarding some candidates.

Logistic Regression				
	TP	TN	FP	FN
A	1	203	57	7
B	3	191	99	5
C	1	286	19	6
D	1	290	16	7
E	8	20	404	0

K-Neighbors Classifier				
	TP	TN	FP	FN
A	1	255	5	7
B	0	290	0	8
C	0	305	0	7
D	0	306	0	8
E	0	423	1	8

Decision Tree Classifier				
	TP	TN	FP	FN
A	1	237	23	7
B	0	290	0	8
C	0	305	0	7
D	0	282	24	8
E	8	121	303	0

Random Forest Classifier				
	TP	TN	FP	FN
A	1	220	40	7
B	2	215	75	6
C	0	305	0	7
D	0	282	24	8
E	8	121	303	0

Figure 5.7: Results Predicting 15 Seconds After

Moreover, for the experiments made with a prediction of 15 seconds before the event happens, the results obtained got, again, a lower performance when compared to the previous one, especially when it comes to the confusion matrix obtained in the test data where almost none case of fall event was detected.

From these results, it is clear that some algorithms, such as Logistic Regression, Decision Tree and Random Forest obtained a reasonable result in detecting the fall events and also got a significant number of False Positives. Moreover, some tests got almost every event of fall right but at the cost of displaying a false alarm as a False Positive. The outcomes of this first experiment made searching for more specific algorithms to deal with class imbalance the main focus of the second experiment.

5.2.2 Second Experiment

The insufficient results from the first round of experiments motivated us to identify new algorithms designed to deal with imbalanced datasets. These algorithms are from the imblearn python library, a variation from the same organization responsible for scikit-learn. Moreover, as cited in one related work, tools for balancing information were needed. For this purpose, experiments were performed with several specific algorithms to deal with imbalanced data. No hyperparameter boosting with Grid Search CV was performed since this experiment aimed to discover the best classifier and balancing algorithm in their "raw" version. Later, we will enhance its performance through hyperparameter boosting in further experiments. Moreover, the pipeline used in this experiment is constituted of one algorithm for balancing, one algorithm for scaling, that is, the Robust Scaler, and the classifier itself.

Figures 5.8, 5.9, 5.10, and 5.11 show the results obtained during this experiment for the train data. The first, second, and third numbers in each cell represent the result for 5, 10, and 15 seconds, respectively.

Balanced Random Forest Classifier - SMOTE					
	TP	RECALL	PRECISION	F1	BALANCED ACCURACY
A	86.8 | 78.4 | 77.6	70 | 68.3 | 70	10.6 | 7.7 | 7.3	18.2 | 13.8 | 13.3	76.4 | 71.1 | 71.6
B	84.7 | 66.1 | 67	65 | 38.3 | 51.7	8.7 | 2.8 | 3.2	15.1 | 5.2 | 5.9	73.5 | 53.7 | 58.9
C	86 | 78.3 | 78.8	55 | 48.3 | 63.3	10.7 | 6.8 | 6	17.7 | 11.9 | 10.9	72 | 66.5 | 70.5
D	70.5 | 78.2 | 74	55 | 65 | 55	9.2 | 8.6 | 5.9	15.5 | 15.2 | 10.6	69.8 | 72.8 | 66.5
E	83.8 | 69.1 | 70.7	68.3 | 38.9 | 50	7.1 | 3.7 | 3.7	12.7 | 6.8 | 6.9	76 | 60.2 | 63.2

Balanced Random Forest Classifier - ADASYN					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	86.5 | 78.9 | 77.4	70 | 68.3 | 75	11.1 | 7.8 | 7.8	18.9 | 13.9 | 14.1	76.8 | 71.2 | 74.1
B	84.7 | 67.9 | 67.3	71.7 | 28.3 | 56.9	9.5 | 1.9 | 3.6	16.6 | 3.6 | 6.7	77 | 49 | 61.2
C	85.8 | 77.6 | 78.9	55 | 53.3 | 63.3	11 | 7.1 | 6	18.2 | 12.6 | 11	72.2 | 68.4 | 70.9
D	70.8 | 77.1 | 73.2	55 | 70 | 50	8.2 | 8.6 | 5.4	14.1 | 15.2 | 9.7	69.1 | 74.4 | 63.8
E	84.9 | 68.7 | 70.4	63.3 | 31.7 | 48.3	7.1 | 2.9 | 3.9	12.7 | 5.3 | 7.2	74.3 | 56.9 | 62.8

Balanced Random Forest Classifier - Borderline SMOTE					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	80.5 | 75.2 | 76.8	46.7 | 45 | 45	9.4 | 5.5 | 10	15.6 | 9.7 | 14.5	67.1 | 62.7 | 65.2
B	83.4 | 68.2 | 62.6	46.7 | 25 | 16.7	11 | 7.2 | 2.3	17.1 | 11.1 | 4	67.8 | 58.8 | 48.2
C	83.5 | 78.8 | 72.9	55 | 55 | 48.3	29.7 | 10.6 | 5.8	17.7 | 10.2	75.3 | 72.1 | 66.3
D	67.7 | 74.5 | 65.8	25 | 45 | 30	7.3 | 8.6 | 4.9	11.2 | 14.4 | 8	58.6 | 66 | 57.3
E	87.6 | 69.1 | 63.7	58.3 | 16.7 | 16.7	7.5 | 4.9 | 1.6	13.1 | 7.1 | 2.9	73 | 53 | 50.2

Balanced Random Forest Classifier - SVMSMOTE					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	81 | 79.3 | 78	26.7 | 13.3 | 10	14.3 | 4.4 | 7.5	18.3 | 6.7 | 8.3	61.3 | 54.7 | 53.2
B	73.4 | 64.7 | 68.3	6.7 | 6.7 | 0	2 | 5 | 0	3.1 | 5.7 | 0	51.4 | 52.2 | 48.1
C	87.1 | 75.8 | 73.4	38.3 | 11.7 | 11.7	45.5 | 7.3 | 7.3	34.1 | 8.9 | 9	68.1 | 54.1 | 54.3
D	77.1 | 81.8 | 74	10 | 15 | 0	8.3 | 9.4 | 0	9 | 11.4 | 0	53.5 | 56.1 | 48.2
E	88.3 | 70.8 | 61.9	23.3 | 0 | 0	28.9 | 0 | 0	24.8 | 0 | 0	60.6 | 48.7 | 48.6

Balanced Random Forest Classifier - SMOTEENN					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	85.9 | 77.7 | 78.6	70 | 63.3 | 75	10.5 | 7 | 7.9	18.1 | 12.6 | 14.2	76.4 | 68.4 | 74.1
B	84.4 | 67.8 | 69.6	76.7 | 33.3 | 56.7	9.1 | 2.7 | 3.7	16.1 | 5 | 6.8	78 | 52.1 | 61.2
C	84.9 | 78.9 | 79.2	60 | 43.3 | 63.3	9.7 | 6.7 | 6.1	16.5 | 11.5 | 11	73.2 | 64.4 | 70.7
D	71.1 | 78.5 | 73.9	45 | 70 | 50	8.3 | 9.1 | 5.2	14 | 16.2 | 9.4	66.9 | 75.1 | 63.8
E	87.3 | 70.4 | 70.5	68.3 | 43.3 | 48.3	7 | 4.3 | 3.7	12.6 | 7.8 | 6.9	75.8 | 63 | 62.2

Balanced Random Forest Classifier - SMOTE Tomek					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	81.3 | 71.1 | 75.9	33.3 | 18.3 | 16.7	12 | 9.7 | 10.2	17.2 | 12.4 | 12.5	64.1 | 57.1 | 56.4
B	74.9 | 61.8 | 70.8	18.3 | 0 | 11.7	7.8 | 0 | 4.9	10.9 | 0 | 6.7	56.5 | 47.8 | 53
C	87.1 | 73 | 72.5	31.7 | 11.7 | 11.7	21.7 | 9 | 6.2	24.4 | 10 | 8	64.2 | 54.2 | 54
D	69.8 | 78.7 | 76.8	5 | 20 | 5	6.7 | 10.8 | 2.2	5.7 | 14 | 3.1	51.2 | 58.2 | 50.3
E	86.2 | 69.6 | 64.4	30 | 5 | 0	19.5 | 5 | 0	22.7 | 5 | 0	63.8 | 50.7 | 48.5

Figure 5.8: Results Obtained from the Second Experiment with Balanced Random Forest

Baggin Classifier - SMOTE

	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	71.4 | 62.6 | 65.2	26.7 | 28.3 | 16.7	10.8 | 17.5 | 8.9	15.1 | 17.8 | 11.5	60.8 | 61 | 55.9
B	62.4 | 58.2 | 57	30 | 0 | 11.7	12.6 | 0 | 3.9	17.4 | 0 | 5.7	62.4 | 47.4 | 52.1
C	74 | 55 | 61.5	26.7 | 6.7 | 21.7	16 | 2.5 | 13.1	19.5 | 3.6 | 15.7	61.5 | 51.5 | 58.6
D	56.9 | 72.1 | 61.6	5 | 15 | 10	5 | 6.5 | 4.3	5 | 9 | 6	51 | 55 | 52.3
E	78.2 | 54 | 54.2	41.7 | 5 | 5	24.3 | 2.2 | 4	29.9 | 3.1 | 4.4	69.4 | 50.3 | 50.4

Baggin Classifier - ADASYN

	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	76.2 | 67.2 | 65.5	26.7 | 23.3 | 21.7	9.4 | 11 | 11.9	13.6 | 14.2 | 15.2	60.6 | 59 | 58.3
B	66.9 | 57.2 | 53.5	23.3 | 13.3 | 11.7	9.6 | 5 | 5.8	13.3 | 7.3 | 7.6	58.8 | 53.5 | 53.3
C	82.1 | 57.2 | 69	31.7 | 6.7 | 16.7	17.3 | 2.5 | 10.2	21.7 | 3.6 | 12.	64 | 51.4 | 56.3
D	64.6 | 65.4 | 58.6	10 | 15 | 10	6.2 | 6.6 | 5.6	7.5 | 8.9 | 7.1	52.8 | 54.8 | 52.6
E	78.2 | 54.3 | 49.1	46.7 | 10 | 0	26.3 | 3.9 | 0	32.4 | 5.6 | 0	71.6 | 52.6 | 47.7

Baggin Classifier - Borderline SMOTE

	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	67.2 | 68.8 | 63.5	26.7 | 6.7 | 5	15 | 5 | 4	18.9 | 5.7 | 4.4	61.5 | 51.8 | 50.5
B	56.5 | 51.1 | 61.5	11.7 | 6.7 | 0	5.6 | 10 | 0	7.3 | 8 | 0	54.1 | 52.1 | 48.3
C	71.8 | 61.1 | 64.6	26.7 | 6.7 | 6.7	33.3 | 5 | 2	23.9 | 5.7 | 3.1	62.6 | 52.1 | 51.3
D	58.6 | 70 | 64.1	5 | 15 | 0	2 | 11.2 | 0	2.9 | 12.6 | 0	50.6 | 56.2 | 48.3
E	68.3 | 59.1 | 50.6	11.7 | 0 | 0	10 | 0 | 0	10.7 | 0 | 0	54.6 | 48.8 | 49.1

Baggin Classifier - SVMSMOTE

	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	68.9 | 66 | 63.4	21.7 | 13.3 | 10	10.5 | 6.7 | 4.4	13.7 | 8.9 | 6.2	58.7 | 54.9 | 52.9
B	71.1 | 58.8 | 66.9	5 | 6.7 | 6.7	2.9 | 5 | 2.5	3.6 | 5.7 | 3.6	50.6 | 52.1 | 51.5
C	73.9 | 62.5 | 60.5	25 | 13.3 | 6.7	31.7 | 9 | 2.9	21.2 | 10.7 | 4	61.4 | 55.1 | 51.4
D	56.8 | 75.8 | 55.2	5 | 20 | 0	3.3 | 12.4 | 0	4 | 15 | 0	50.8 | 58.3 | 47.8
E	70.3 | 63.8 | 56.8	16.7 | 0 | 0	10.6 | 0 | 0	12.5 | 0 | 0	56.8 | 48.5 | 48.1

Baggin Classifier - SMOTEENN

	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	70.2 | 68.6 | 64.2	33.3 | 38.3 | 20	8.2 | 13.9 | 5	12.9 | 19.7 | 8	62.6 | 65.1 | 53.6
B	63 | 57.1 | 63.5	30 | 35 | 20	7.6 | 8.5 | 2.8	12 | 13.6 | 4.9	60.4 | 62.8 | 54
C	77.4 | 58.9 | 64.6	33.3 | 11.7 | 26.7	13.2 | 3.2 | 8.7	18.3 | 4.9 | 12.9	63.5 | 52.1 | 59.5
D	64.4 | 75 | 64.6	15 | 45 | 25	4.5 | 11.6 | 6.4	6.9 | 18.2 | 10.2	53.6 | 67.7 | 58.2
E	72.8 | 53.9 | 54.6	35 | 0 | 5	11.3 | 0 | 1.4	16.5 | 0 | 2.2	64.5 | 46.3 | 48.4

Baggin Classifier - SMOTE Tomek

	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	76.1 | 67.1 | 70.9	31.7 | 23.3 | 15	14.3 | 14.4 | 6.9	19.2 | 16.7 | 9.5	63.3 | 59.3 | 55.1
B	62.1 | 58.2 | 56.4	25 | 0 | 6.7	9.8 | 0 | 1.3	13.7 | 0 | 2.1	59.7 | 46.7 | 49.9
C	75.8 | 57.7 | 66.2	20 | 6.7 | 16.7	8.5 | 2.9 | 9.7	11.9 | 4 | 11.7	58.3 | 51.3 | 55.8
D	52.2 | 72.7 | 58.6	10 | 15 | 10	6.7 | 6.4 | 4.7	8 | 8.9 | 6.4	53.2 | 54.8 | 52.5
E	70.8 | 58.7 | 50.9	41.7 | 5 | 0	16.5 | 2.9 | 0	23 | 3.6 | 0	69.1 | 50.4 | 47.8

Figure 5.9: Results Obtained from the Second Experiment with Balanced Baggin Classifier

RUS Boost - SMOTE					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	74.9 | 73.5 | 71.5	38.3 | 33.3 | 26.7	13.3 | 10.2 | 9.4	18.7 | 14.9 | 13.7	65.7 | 62.1 | 59
B	76.3 | 59 | 62.2	31.7 | 23.3 | 16.7	9.8 | 4.2 | 4.4	14.8 | 7.1 | 6.8	62.1 | 55.9 | 52.6
C	71.6 | 63 | 71.2	31.7 | 28.3 | 26.7	13.4 | 12.2 | 9.6	17.6 | 16 | 13.3	63 | 61.8 | 59.5
D	62.1 | 70.9 | 65.3	20 | 45 | 5	7.4 | 11.8 | 1.8	10.7 | 18.6 | 2.7	56.4 | 68 | 49.2
E	70 | 59.3 | 55.3	35 | 11.7 | 5	12.9 | 1.9 | 0.8	18.8 | 3.3 | 1.4	65.3 | 51.1 | 47.8

RUS Boost - ADASYN					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	73.5 | 74.9 | 69.8	38.3 | 28.3 | 26.7	11.5 | 9.5 | 10.2	17.4 | 13.7 | 14.6	65.7 | 60.6 | 59.9
B	79 | 64.8 | 61.5	31.7 | 28.3 | 16.7	9.2 | 5.7 | 4.1	14.1 | 9.4 | 6.5	62.1 | 57.9 | 52.6
C	69.9 | 69.8 | 74.9	26.7 | 11.7 | 28.3	13.4 | 3.5 | 9.2	17.1 | 5.4 | 13	61 | 53.1 | 60.2
D	65.4 | 68.6 | 68.8	20 | 30 | 15	6.3 | 7 | 4.9	9.4 | 11.3 | 7.4	56.2 | 59.9 | 54.2
E	69.1 | 56.5 | 58.3	35 | 6.7 | 0	11.8 | 0.7 | 0	17 | 1.3 | 0	65.1 | 48.2 | 45.9

RUS Boost - Borderline SMOTE					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	76.7 | 75.5 | 73.1	21.7 | 28.3 | 16.7	8.6 | 8.7 | 6.9	12.2 | 13.1 | 9.5	57.8 | 60.8 | 55.5
B	79.3 | 63.4 | 55.3	11.7 | 11.7 | 11.7	4.3 | 5.6 | 4.4	6.2 | 7.3 | 5.9	53.1 | 53.4 | 51.9
C	69.8 | 63.6 | 72.9	20 | 16.7 | 16.7	16.2 | 7.8 | 7.8	15.3 | 10.6 | 10	57.8 | 56.2 | 55.1
D	63.8 | 70.1 | 63.7	20 | 30 | 10	10.7 | 12 | 4	13.1 | 16.9 | 5.7	57.3 | 62.1 | 51.9
E	79.1 | 53.8 | 58.1	30 | 6.7 | 0	10.9 | 4 | 0	15.4 | 5 | 0	62.9 | 51.2 | 47

RUS Boost - SVMSMOTE					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	76.7 | 75.4 | 73.2	31.7 | 23.3 | 28.3	12.9 | 7.3 | 11.6	18 | 11 | 16.2	63.3 | 58.6 | 61.4
B	76.2 | 61.2 | 64.9	18.3 | 11.7 | 11.7	9.2 | 5.4 | 4.5	11.7 | 7.3 | 6	56.3 | 53.5 | 51.8
C	69.5 | 66.5 | 74.8	25 | 18.3 | 21.7	15.7 | 7.8 | 8.1	17 | 10.9 | 11.4	60.5 | 57 | 57.1
D	73.2 | 68 | 64.2	25 | 25 | 5	10.7 | 10.3 | 2	15 | 14.5 | 2.9	60.1 | 59.6 | 49.5
E	81.9 | 50.5 | 60.4	36.7 | 0 | 0	36.7 | 0 | 0	18 | 0 | 0	23.1 | 47.4 | 46.6

RUS Boost - SMOTEENN					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	76.5 | 75.8 | 69.5	31.7 | 38.3 | 36.7	8.1 | 8.9 | 10.3	12.7 | 14.2 | 15.8	61.3 | 63.5 | 62.5
B	77.3 | 61.3 | 64.5	41.7 | 23.3 | 30	9.4 | 3.8 | 4.5	15.2 | 6.4 | 7.8	65.8 | 54.5 | 57.1
C	72.7 | 72.6 | 72	36.7 | 26.7 | 26.7	11.6 | 6.3 | 6	16.8 | 10.2 | 9.7	64.2 | 59.4 | 57.6
D	64.2 | 73.7 | 68.9	35 | 50 | 25	9 | 9.9 | 5.7	14.1 | 16.4 | 9.2	62.3 | 68.8 | 57.4
E	80.2 | 59.5 | 59.7	45 | 11.7 | 0	9.8 | 1.4 | 0	15.9 | 2.5 | 0	68.9 | 49.6 | 44.8

RUS Boost - SMOTE Tomek					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	74.9 | 73.5 | 71.8	38.3 | 33.3 | 21.7	13.3 | 10.4 | 6.8	18.7 | 15.2 | 10.1	65.7 | 62.2 | 56.5
B	75.1 | 59.8 | 59.8	31.7 | 18.3 | 16.7	9.3 | 3 | 4.5	14.3 | 5.1 | 6.9	62 | 53.1 | 52.7
C	71.5 | 63 | 71.2	31.7 | 28.3 | 26.7	13.4 | 12.2 | 9.3	17.6 | 16 | 13	62.8 | 61.8 | 59.4
D	62.2 | 69.8 | 62.2	20 | 35 | 5	7.4 | 9.7 | 1.8	10.7 | 15.1 | 2.7	56.4 | 62.6 | 49
E	68 | 59.3 | 57.5	35 | 11.7 | 5	12.9 | 1.9 | 0.8	18.8 | 3.3 | 1.4	65.5 | 51.1 | 47.9

Figure 5.10: Results Obtained from the Second Experiment with RUS Boost Classifier

Easy Ensemble - SMOTE					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	74.9 | 73.5 | 71.5	38.3 | 33.3 | 26.7	13.3 | 10.2 | 9.4	18.7 | 14.9 | 13.7	65.7 | 62.1 | 59
B	76.3 | 59 | 62.2	31.7 | 23.3 | 16.7	9.8 | 4.2 | 4.4	14.8 | 7.1 | 6.8	62.1 | 55.9 | 52.6
C	71.6 | 63 | 71.2	31.7 | 28.3 | 26.7	13.4 | 12.2 | 9.6	17.6 | 16 | 13.3	63 | 61.8 | 59.5
D	62.2 | 70.9 | 65.3	20 | 45 | 5	7.4 | 11.8 | 1.8	10.7 | 18.6 | 2.7	56.4 | 68 | 49.2
E	70 | 59.3 | 55.3	35 | 11.7 | 5	12.9 | 1.9 | 0.8	18.8 | 3.3 | 1.4	65.3 | 51.1 | 47.8

Easy Ensemble - ADASYN					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	72.6 | 74.6 | 71.1	33.3 | 28.3 | 21.7	9.1 | 9.8 | 8.7	14.1 | 14.2 | 12.3	62.9 | 60.6 | 57.4
B	78.6 | 63.3 | 61.1	25 | 16.7 | 16.7	8.2 | 3.6 | 4.6	12.3 | 5.9 | 7.1	59.1 | 52.8 | 52.9
C	70 | 65.9 | 73.6	26.7 | 11.7 | 33.3	14.3 | 3.9 | 11	18.1 | 5.9 | 15.5	61.2 | 53.4 | 62.7
D	65.8 | 68.2 | 66.3	20 | 30 | 5	6 | 8.2 | 1.7	9.2 | 12.9 | 2.5	56.1 | 60.8 | 48.7
E	70.7 | 57.8 | 60.3	40 | 11.7 | 0	14 | 1.79 | 0	20.1 | 3 | 0	67.8 | 50.9 | 46.1

Easy Ensemble - Borderline SMOTE					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	76.7 | 75.5 | 73.1	21.7 | 28.3 | 16.7	8.6 | 8.7 | 6.9	12.2 | 13.1 | 9.5	57.8 | 60.8 | 55.5
B	79.3 | 63.4 | 55.3	11.7 | 11.7 | 11.7	4.3 | 5.6 | 4.4	6.2 | 7.3 | 5.9	53.1 | 53.4 | 51.9
C	69.8 | 63.6 | 72.9	20 | 16.7 | 16.7	16.2 | 7.8 | 7.8	15.3 | 10.6 | 10	57.8 | 56.2 | 55.1
D	63.8 | 70.1 | 63.7	20 | 30 | 10	10.7 | 12 | 4	13.1 | 16.9 | 5.7	57.3 | 62.1 | 51.9
E	79.1 | 53.8 | 58.1	30 | 6.7 | 0	10.9 | 4 | 0	15.4 | 5 | 0	62.9 | 51.2 | 47

Easy Ensemble - SVMSMOTE					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	77.8 | 75.6 | 76.1	36.7 | 28.3 | 16.7	13.6 | 10 | 8.1	19.7 | 14.5 | 10.7	65.4 | 61 | 55.9
B	80.9 | 63.3 | 60.3	11.7 | 6.7 | 11.7	4.3 | 2 | 5.1	6.2 | 3.1 | 6.3	52.9 | 51.3 | 51.8
C	73.5 | 65.8 | 75.1	20 | 16.7 | 21.7	17.3 | 8 | 9.8	16.4 | 10.7 | 12.8	58.3 | 56.2 | 57.8
D	63.4 | 73.7 | 72.4	30 | 30 | 10	13.3 | 11.9 | 4.2	18.1 | 16.7 | 5.9	62.1 | 62 | 52
E	79 | 56.4 | 60.2	35 | 0 | 0	15 | 0 | 0	20.8 | 0 | 0	65.6 | 48.2 | 47.3

Easy Ensemble - SMOTEENN					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	73.9 | 71.7 | 70.4	26.7 | 33.3 | 31.7	6.9 | 8.1 | 9.8	10.8 | 12.8 | 14.7	59.3 | 61.3 | 60.2
B	78.7 | 60.1 | 63	35 | 23.3 | 30	9.1 | 3.9 | 4.9	14.3 | 6.6 | 8.3	62.9 | 55.2 | 57.6
C	70.4 | 64.7 | 71.4	31.7 | 21.7 | 26.7	11.4 | 5.9 | 6.6	16.1 | 9.3 | 10.6	62.1 | 57.1 | 58.8
D	66.8 | 71.5 | 72.3	40 | 45 | 25	11.6 | 8.7 | 6.3	17.8 | 14.5 | 9.9	65.4 | 66.3 | 57.9
E	80.6 | 54 | 64.7	46.7 | 5 | 5	11.3 | 0.8 | 1.3	17.9 | 1.3 | 2	70.2 | 46.5 | 47.5

Easy Ensemble - SMOTE Tomek					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	74.9 | 73.5 | 71.8	38.3 | 33.3 | 21.7	13.3 | 10.4 | 6.8	18.7 | 15.2 | 10.1	65.7 | 62.2 | 56.4
B	75.1 | 59.8 | 59.8	31.7 | 18.3 | 16.7	9.3 | 3 | 4.5	14.3 | 5.1 | 6.9	62 | 53.1 | 52.7
C	71.5 | 63 | 71.2	31.7 | 28.3 | 26.7	13.4 | 12.2 | 9.3	17.6 | 16 | 13	62.8 | 61.8 | 59.4
D	62.2 | 69.8 | 62.2	20 | 35 | 5	7.4 | 9.7 | 1.8	10.7 | 15.1 | 2.7	56.4 | 62.6 | 49
E	68 | 59.3 | 57.5	35 | 11.7 | 5	12.9 | 1.9 | 0.8	18.8 | 3.3 | 1.4	65.5 | 51.1 | 47.9

Figure 5.11: Results Obtained from the Second Experiment with Easy Ensemble Classifier

Figures 5.12, 5.13, 5.14, 5.15, 5.16 show the results considering a prediction with a time interval of 5 seconds before the events happen. In this experiment, a similar observation regarding Recall, Precision, F1, and Balanced Accuracy can be compared to the first experiment's results. Similar results were obtained through all the classifiers and balancing techniques, and none with a very high prominence to the point of positively distancing itself from the result of the others.

Candidate A

	Balanced Random Forest	Balanced Bagging Classifier	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 5 TN: 208 FP: 55 FN: 3	TP: 3 TN: 250 FP: 13 FN: 5	TP: 4 TN: 235 FP: 28 FN: 4	TP: 4 TN: 235 FP: 28 FN: 4
ADASYN	TP: 4 TN: 211 FP: 52 FN: 4	TP: 3 TN: 248 FP: 15 FN: 5	TP: 4 TN: 238 FP: 25 FN: 4	TP: 4 TN: 235 FP: 28 FN: 4
Borderline SMOTE	TP: 4 TN: 216 FP: 47 FN: 4	TP: 0 TN: 253 FP: 10 FN: 8	TP: 1 TN: 247 FP: 16 FN: 7	TP: 1 TN: 247 FP: 16 FN: 7
SVMSMOTE	TP: 2 TN: 248 FP: 15 FN: 6	TP: 0 TN: 250 FP: 13 FN: 8	TP: 3 TN: 247 FP: 16 FN: 5	TP: 4 TN: 238 FP: 25 FN: 4
SMOTEENN	TP:.6 TN:.147 FP:.116 FN: 2	TP: 4 TN: 229 FP: 34 FN: 4	TP: 4 TN: 234 FP: 29 FN: 4	TP: 4 TN: 232 FP: 31 FN: 4
SMOTETomek	TP: 3 TN: 251 FP: 12 FN: 5	TP: 3 TN: 242 FP: 21 FN: 5	TP: 4 TN: 235 FP: 28 FN: 4	TP: 4 TN: 235 FP: 28 FN: 4

Figure 5.12: Results Predicting 5 Seconds After for Candidate A

Candidate B

	Balanced Random Forest	Balanced Bagging Classifier	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 5 TN: 248 FP: 45 FN: 3	TP: 2 TN: 279 FP: 14 FN: 6	TP: 3 TN: 270 FP: 23 FN: 5	TP: 3 TN: 270 FP: 23 FN: 5
ADASYN	TP: 5 TN: 247 FP: 46 FN: 3	TP: 2 TN: 281 FP: 12 FN: 6	TP: 5 TN: 267 FP: 26 FN: 3	TP: 3 TN: 270 FP: 23 FN: 5
Borderline SMOTE	TP: 5 TN: 262 FP: 31 FN: 3	TP: 0 TN: 288 FP: 5 FN: 8	TP: 2 TN: 276 FP: 17 FN: 6	TP: 2 TN: 276 FP: 17 FN: 6
SVMSMOTE	TP: 1 TN: 285 FP: 8 FN: 7	TP: 0 TN: 285 FP: 8 FN: 8	TP: 1 TN: 278 FP: 15 FN: 7	TP: 3 TN: 279 FP: 14 FN: 5
SMOTEENN	TP: 7 TN: 162 FP: 131 FN: 1	TP: 4 TN: 271 FP: 22 FN: 4	TP: 4 TN: 268 FP: 25 FN: 4	TP: 5 TN: 266 FP: 27 FN: 3
SMOTETomek	TP: 2 TN: 281 FP: 12 FN: 6	TP: 4 TN: 279 FP: 14 FN: 4	TP: 3 TN: 270 FP: 23 FN: 5	TP: 3 TN: 270 FP: 23 FN: 5

Figure 5.13: Results Predicting 5 Seconds After for Candidate B

Candidate C				
	Balanced Random Forest	Balanced Bagging Classifier	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 2 TN: 276 FP: 32 FN: 5	TP: 1 TN: 298 FP: 10 FN: 6	TP: 2 TN: 286 FP: 22 FN: 5	TP: 2 TN: 286 FP: 22 FN: 5
ADASYN	TP: 2 TN: 272 FP: 36 FN: 5	TP: 1 TN: 295 FP: 13 FN: 6	TP: 2 TN: 288 FP: 20 FN: 5	TP: 2 TN: 288 FP: 20 FN: 5
Borderline SMOTE	TP: 1 TN: 290 FP: 18 FN: 6	TP: 1 TN: 304 FP: 4 FN: 6	TP: 1 TN: 295 FP: 13 FN: 6	TP: 1 TN: 295 FP: 13 FN: 6
SVMSMOTE	TP: 1 TN: 305 FP: 3 FN: 6	TP: 1 TN: 303 FP: 5 FN: 6	TP: 1 TN: 296 FP: 12 FN: 6	TP: 1 TN: 299 FP: 9 FN: 6
SMOTEENN	TP: 7 TN: 183 FP: 125 FN: 0	TP: 2 TN: 289 FP: 19 FN: 5	TP: 2 TN: 278 FP: 30 FN: 5	TP: 3 TN: 282 FP: 26 FN: 4
SMOTETomek	TP: 1 TN: 303 FP: 5 FN: 6	TP: 1 TN: 294 FP: 14 FN: 6	TP: 1 TN: 289 FP: 19 FN: 6	TP: 1 TN: 289 FP: 19 FN: 6

Figure 5.14: Results Predicting 5 Seconds After for Candidate C

Candidate D				
	Balanced Random Forest	Balanced Bagging Classifie	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 6 TN: 247 FP: 62 FN: 2	TP: 0 TN: 295 FP: 14 FN: 8	TP: 2 TN: 278 FP: 31 FN: 6	TP: 2 TN: 278 FP: 31 FN: 6
ADASYN	TP: 6 TN: 255 FP: 54 FN: 2	TP: 0 TN: 296 FP: 13 FN: 8	TP: 2 TN: 283 FP: 26 FN: 6	TP: 2 TN: 281 FP: 28 FN: 6
Borderline SMOTE	TP: 2 TN: 285 FP: 24 FN: 6	TP: 0 TN: 297 FP: 12 FN: 8	TP: 0 TN: 290 FP: 19 FN: 8	TP: 0 TN: 290 FP: 19 FN: 8
SVMSMOTE	TP: 0 TN: 299 FP: 10 FN: 8	TP: 0 TN: 301 FP: 8 FN: 8	TP: 1 TN: 291 FP: 18 FN: 7	TP: 1 TN: 287 FP: 22 FN: 7
SMOTEENN	TP: 8 TN: 139 FP: 170 FN: 0	TP: 2 TN: 284 FP: 25 FN: 6	TP: 3 TN: 272 FP: 37 FN: 5	TP: 3 TN: 279 FP: 30 FN: 5
SMOTETomek	TP: 0 TN: 298 FP: 11 FN: 8	TP: 0 TN: 296 FP: 13 FN: 8	TP: 2 TN: 278 FP: 31 FN: 6	TP: 2 TN: 278 FP: 31 FN: 6

Figure 5.15: Results Predicting 5 Seconds After for Candidate D

Candidate E				
	Balanced Random Forest	Balanced Bagging Classifier	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 8 TN: 348 FP: 79 FN: 0	TP: 1 TN: 411 FP: 16 FN: 7	TP: 2 TN: 401 FP: 26 FN: 6	TP: 2 TN: 401 FP: 26 FN: 6
ADASYN	TP: 8 TN: 359 FP: 68 FN: 0	TP: 2 TN: 417 FP: 10 FN: 6	TP: 2 TN: 404 FP: 23 FN: 6	TP: 2 TN: 406 FP: 21 FN: 6
Borderline SMOTE	TP: 4 TN: 382 FP: 45 FN: 4	TP: 2 TN: 415 FP: 12 FN: 6	TP: 3 TN: 409 FP: 18 FN: 5	TP: 3 TN: 409 FP: 18 FN: 5
SVMSMOTE	TP: 1 TN: 416 FP: 11 FN: 7	TP: 3 TN: 411 FP: 16 FN: 5	TP: 2 TN: 409 FP: 18 FN: 6	TP: 2 TN: 405 FP: 22 FN: 6
SMOTEENN	TP: 8 TN: 262 FP: 165 FN: 0	TP: 3 TN: 396 FP: 31 FN: 5	TP: 4 TN: 390 FP: 37 FN: 4	TP: 3 TN: 399 FP: 28 FN: 5
SMOTETomek	TP: 3 TN: 416 FP: 11 FN: 5	TP: 2 TN: 411 FP: 16 FN: 6	TP: 2 TN: 401 FP: 26 FN: 6	TP: 2 TN: 401 FP: 26 FN: 6

Figure 5.16: Results Predicting 5 Seconds After for Candidate E

Figures 5.17, 5.18, 5.19, 5.20, 5.21 show the results considering a time interval of 10 seconds before the events happen. In this second round of observations, two algorithms can be discarded regarding their poor results and lack of performance, pushing the prediction time a little bit further. They are the *SVMSMOTE* and *Borderline SMOTE* algorithms.

Candidate A

	Balanced Random Forest	Balanced Bagging Classifier	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 6 TN: 185 FP: 77 FN: 2	TP: 1 TN: 246 FP: 16 FN: 7	TP: 2 TN: 244 FP: 18 FN: 6	TP: 2 TN: 244 FP: 18 FN: 6
ADASYN	TP: 6 TN: 186 FP: 76 FN: 2	TP: 2 TN: 243 FP: 19 FN: 6	TP: 3 TN: 241 FP: 21 FN: 5	TP: 3 TN: 241 FP: 21 FN: 5
Borderline SMOTE	TP: 5 TN: 195 FP: 67 FN: 3	TP: 0 TN: 246 FP: 16 FN: 8	TP: 3 TN: 243 FP: 19 FN: 5	TP: 3 TN: 243 FP: 19 FN: 5
SVMSMOTE	TP: 2 TN: 250 FP: 12 FN: 6	TP: 2 TN: 246 FP: 16 FN: 6	TP: 2 TN: 245 FP: 17 FN: 6	TP: 2 TN: 268 FP: 24 FN: 6
SMOTEENN	TP: 8 TN: 124 FP: 138 FN: 0	TP: 2 TN: 237 FP: 25 FN: 6	TP: 2 TN: 230 FP: 32 FN: 6	TP: 2 TN: 238 FP: 24 FN: 6
SMOTETomek	TP: 0 TN: 251 FP: 11 FN: 8	TP: 1 TN: 245 FP: 17 FN: 7	TP: 2 TN: 244 FP: 18 FN: 6	TP: 2 TN: 244 FP: 18 FN: 6

Figure 5.17: Results Predicting 10 Seconds After for Candidate A

Candidate B

	Balanced Random Forest	Balanced Bagging Classifier	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 4 TN: 177 FP: 115 FN: 4	TP: 2 TN: 274 FP: 18 FN: 6	TP: 2 TN: 250 FP: 42 FN: 6	TP: 2 TN: 250 FP: 42 FN: 6
ADASYN	TP: 5 TN: 177 FP: 115 FN: 3	TP: 2 TN: 274 FP: 18 FN: 6	TP: 2 TN: 250 FP: 42 FN: 6	TP: 2 TN: 246 FP: 46 FN: 6
Borderline SMOTE	TP: 3 TN: 259 FP: 33 FN: 5	TP: 2 TN: 275 FP: 17 FN: 6	TP: 2 TN: 271 FP: 21 FN: 6	TP: 2 TN: 271 FP: 21 FN: 6
SVMSMOTE	TP: 2 TN: 278 FP: 14 FN: 6	TP: 2 TN: 278 FP: 14 FN: 6	TP: 3 TN: 272 FP: 20 FN: 5	TP: 0 TN: 296 FP: 10 FN: 8
SMOTEENN	TP: 8 TN: 134 FP: 158 FN: 0	TP: 2 TN: 252 FP: 40 FN: 6	TP: 2 TN: 248 FP: 44 FN: 6	TP: 2 TN: 247 FP: 45 FN: 6
SMOTETomek	TP: 2 TN: 275 FP: 17 FN: 6	TP: 2 TN: 270 FP: 22 FN: 6	TP: 2 TN: 249 FP: 43 FN: 6	TP: 2 TN: 249 FP: 43 FN: 6

Figure 5.18: Results Predicting 10 Seconds After for Candidate B

Candidate C				
	Balanced Random Forest	Balanced Bagging Classifie	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 3 TN: 257 FP: 49 FN: 5	TP: 2 TN: 291 FP: 15 FN: 6	TP: 4 TN: 286 FP: 20 FN: 4	TP: 4 TN: 286 FP: 20 FN: 4
ADASYN	TP: 3 TN: 258 FP: 48 FN: 5	TP: 2 TN: 294 FP: 12 FN: 6	TP: 3 TN: 287 FP: 19 FN: 5	TP: 3 TN: 282 FP: 24 FN: 5
Borderline SMOTE	TP: 1 TN: 284 FP: 22 FN: 7	TP: 0 TN: 302 FP: 4 FN: 8	TP: 0 TN: 293 FP: 13 FN: 8	TP: 0 TN: 293 FP: 13 FN: 8
SVMSMOTE	TP: 0 TN: 299 FP: 7 FN: 8	TP: 0 TN: 299 FP: 7 FN: 8	TP: 0 TN: 293 FP: 13 FN: 8	TP: 0 TN: 296 FP: 10 FN: 8
SMOTEENN	TP: 8 TN: 171 FP: 135 FN: 0	TP: 2 TN: 283 FP: 23 FN: 6	TP: 4 TN: 284 FP: 22 FN: 4	TP: 4 TN: 287 FP: 19 FN: 4
SMOTETomek	TP: 2 TN: 297 FP: 9 FN: 6	TP: 2 TN: 290 FP: 16 FN: 6	TP: 4 TN: 286 FP: 20 FN: 4	TP: 4 TN: 286 FP: 20 FN: 4

Figure 5.19: Results Predicting 10 Seconds After for Candidate C

Candidate D				
	Balanced Random Forest	Balanced Bagging Classifier	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 3 TN: 247 FP: 60 FN: 5	TP: 0 TN: 293 FP: 14 FN: 8	TP: 1 TN: 282 FP: 25 FN: 7	TP: 1 TN: 282 FP: 25 FN: 7
ADASYN	TP: 4 TN: 248 FP: 59 FN: 4	TP: 0 TN: 294 FP: 13 FN: 8	TP: 1 TN: 189 FP: 18 FN: 7	TP: 1 TN: 287 FP: 20 FN: 7
Borderline SMOTE	TP: 1 TN: 268 FP: 39 FN: 7	TP: 0 TN: 299 FP: 8 FN: 8	TP: 1 TN: 213 FP: 16 FN: 7	TP: 1 TN: 291 FP: 16 FN: 7
SVMSMOTE	TP: 0 TN: 299 FP: 8 FN: 8	TP: 0 TN: 297 FP: 10 FN: 8	TP: 2 TN: 295 FP:12 FN: 6	TP: 1 TN: 289 FP: 18 FN: 7
SMOTEENN	TP: 6 TN: 157 FP: 150 FN: 2	TP: 0 TN: 284 FP: 23 FN: 8	TP: 2 TN: 277 FP: 30 FN: 6	TP: 1 TN: 283 FP: 24 FN: 7
SMOTETomek	TP: 0 TN: 296 FP: 11 FN: 8	TP: 1 TN: 295 FP: 12 FN: 7	TP: 1 TN: 282 FP: 25 FN: 7	TP: 1 TN: 282 FP: 25 FN: 7

Figure 5.20: Results Predicting 10 Seconds After for Candidate D

Candidate E				
	Balanced Random Forest	Balanced Bagging Classifier	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 4 TN: 364 FP: 62 FN: 4	TP: 0 TN: 412 FP: 14 FN: 8	TP: 3 TN: 377 FP: 49 FN: 5	TP: 3 TN: 377 FP: 49 FN: 5
ADASYN	TP: 4 TN: 373 FP: 53 FN: 4	TP: 1 TN: 406 FP: 20 FN: 7	TP: 2 TN: 373 FP: 53 FN: 6	TP: 2 TN: 373 FP: 53 FN: 6
Borderline SMOTE	TP: 4 TN: 342 FP: 84 FN: 4	TP: 0 TN: 413 FP: 13 FN: 8	TP: 2 TN: 393 FP: 33 FN: 6	TP: 2 TN: 393 FP: 33 FN: 6
SVMSMOTE	TP: 0 TN: 414 FP: 12 FN: 8	TP: 0 TN: 418 FP: 8 FN: 8	TP: 2 TN: 395 FP: 31 FN: 6	TP: 1 TN: 396 FP: 30 FN: 7
SMOTEENN	TP: 5 TN: 150 FP: 276 FN: 3	TP: 2 TN: 395 FP: 31 FN: 6	TP: 2 TN: 367 FP: 59 FN: 6	TP: 2 TN: 372 FP: 54 FN: 6
SMOTETomek	TP: 0 TN: 415 FP: 11 FN: 8	TP: 0 TN: 410 FP: 16 FN: 8	TP: 3 TN: 377 FP: 49 FN: 5	TP: 3 TN: 377 FP: 49 FN: 5

Figure 5.21: Results Predicting 10 Seconds After for Candidate E

Figures 5.22, 5.23, 5.24, 5.25, and 5.26 summarize the results considering 15 seconds before the events happen. Since the goal is to detect as many fall events as possible due to the health information we are dealing with, one False Positive might result in deep problems. Therefore, the best algorithm and technique is the one that points correctly to most of the occurrences of True Positives with a lower rate of False Positives. This way, the best results came from the Balanced Random Forest Algorithm with the SMOTEENN balancing technique. Thus, more experiments with these two tools to deal with imbalanced data were chosen to be deeply tested in the next experiment.

Candidate A				
	Balanced Random Forest	Balanced Bagging Classifier	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 4 TN: 183 FP: 77 FN: 4	TP: 1 TN: 244 FP: 16 FN: 7	TP: 2 TN: 236 FP: 24 FN: 6	TP: 2 TN: 236 FP: 24 FN: 6
ADASYN	TP: 4 TN: 179 FP: 81 FN: 4	TP: 1 TN: 250 FP: 10 FN: 7	TP: 2 TN: 235 FP: 25 FN: 6	TP: 2 TN: 235 FP: 25 FN: 6
Borderline SMOTE	TP: 2 TN: 196 FP: 64 FN: 6	TP: 1 TN: 245 FP: 15 FN: 7	TP: 2 TN: 249 FP: 11 FN: 6	TP: 2 TN: 249 FP: 11 FN: 6
SVMSMOTE	TP: 1 TN: 249 FP: 11 FN: 7	TP: 1 TN: 240 FP: 20 FN: 7	TP: 0 TN: 249 FP: 11 FN: 8	TP: 1 TN: 245 FP: 15 FN: 7
SMOTEENN	TP: 8 TN: 125 FP: 135 FN: 0	TP: 2 TN: 231 FP: 29 FN: 6	TP: 1 TN: 231 FP: 29 FN: 7	TP: 2 TN: 233 FP: 27 FN: 6
SMOTETomek	TP: 1 TN: 251 FP: 9 FN: 7	TP: 1 TN: 240 FP: 20 FN: 7	TP: 3 TN: 230 FP: 30 FN: 5	TP: 3 TN: 230 FP: 30 FN: 5

Figure 5.22: Results Predicting 15 Seconds After for Candidate A

Candidate B				
	Balanced Random Forest	Balanced Bagging Classifie	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 4 TN: 177 FP: 113 FN: 4	TP: 2 TN: 280 FP: 10 FN: 6	TP: 1 TN: 263 FP: 27 FN: 7	TP: 1 TN: 263 FP: 27 FN: 7
ADASYN	TP: 4 TN: 183 FP: 105 FN: 4	TP: 2 TN: 279 FP: 11 FN: 6	TP: 2 TN: 259 FP: 31 FN: 6	TP: 2 TN: 264 FP: 26 FN: 6
Borderline SMOTE	TP: 4 TN: 189 FP: 101 FN: 4	TP: 1 TN: 282 FP: 8 FN: 7	TP: 2 TN: 268 FP: 22 FN: 6	TP: 2 TN: 268 FP: 22 FN: 6
SVMSMOTE	TP: 2 TN: 282 FP: 8 FN: 6	TP: 1 TN: 274 FP: 16 FN: 7	TP: 1 TN: 274 FP: 16 FN: 7	TP: 2 TN: 269 FP: 21 FN: 6
SMOTEENN	TP: 8 TN: 122 FP: 168 FN: 0	TP: 2 TN: 265 FP: 25 FN: 6	TP: 2 TN: 262 FP: 28 FN: 6	TP: 2 TN: 263 FP: 27 FN: 6
SMOTETomek	TP: 1 TN: 282 FP: 8 FN: 7	TP: 1 TN: 281 FP: 9 FN: 7	TP: 2 TN: 263 FP: 27 FN: 6	TP: 2 TN: 263 FP: 27 FN: 6

Figure 5.23: Results Predicting 15 Seconds After for Candidate B

Candidate C				
	Balanced Random Forest	Balanced Bagging Classifier	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 5 TN: 238 FP: 67 FN: 2	TP: 1 TN: 297 FP: 8 FN: 6	TP: 3 TN: 290 FP: 15 FN: 4	TP: 3 TN: 290 FP: 15 FN: 4
ADASYN	TP: 4 TN: 239 FP: 66 FN: 3	TP: 2 TN: 296 FP: 9 FN: 5	TP: 2 TN: 288 FP: 17 FN: 5	TP: 2 TN: 289 FP: 16 FN: 5
Borderline SMOTE	TP: 2 TN: 263 FP: 42 FN: 5	TP: 2 TN: 293 FP: 12 FN: 5	TP: 1 TN: 292 FP: 13 FN: 6	TP: 1 TN: 292 FP: 13 FN: 6
SVMSMOTE	TP: 2 TN: 299 FP: 6 FN: 5	TP: 2 TN: 295 FP: 10 FN: 5	TP: 1 TN: 292 FP: 13 FN: 6	TP: 2 TN: 295 FP: 10 FN: 5
SMOTEENN	TP: 7 TN: 170 FP: 135 FN: 0	TP: 3 TN: 283 FP: 22 FN: 4	TP: 2 TN: 280 FP: 25 FN: 5	TP: 4 TN: 285 FP: 20 FN: 3
SMOTETomek	TP: 2 TN: 300 FP: 5 FN: 5	TP: 2 TN: 294 FP: 11 FN: 5	TP: 3 TN: 290 FP: 15 FN: 4	TP: 3 TN: 290 FP: 15 FN: 4

Figure 5.24: Results Predicting 15 Seconds After for Candidate C

Candidate D				
	Balanced Random Forest	Balanced Bagging Classifier	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 4 TN: 237 FP: 69 FN: 4	TP: 1 TN: 291 FP: 15 FN: 7	TP: 2 TN: 280 FP: 26 FN: 6	TP: 2 TN: 280 FP: 26 FN: 6
ADASYN	TP: 4 TN: 231 FP: 75 FN: 4	TP: 1 TN: 292 FP: 14 FN: 7	TP: 0 TN: 283 FP: 23 FN: 8	TP: 2 TN: 284 FP: 22 FN: 6
Borderline SMOTE	TP: 3 TN: 237 FP: 69 FN: 5	TP: 0 TN: 292 FP: 14 FN: 8	TP: 1 TN: 282 FP: 24 FN: 7	TP: 1 TN: 282 FP: 24 FN: 7
SVMSMOTE	TP: 0 TN: 290 FP: 16 FN: 8	TP: 1 TN: 289 FP: 17 FN: 7	TP: 2 TN: 279 FP: 27 FN: 6	TP: 0 TN: 287 FP: 19 FN: 8
SMOTEENN	TP: 8 TN: 171 FP: 135 FN: 0	TP: 2 TN: 281 FP: 25 FN: 6	TP: 1 TN: 277 FP: 29 FN: 7	TP: 1 TN: 281 FP: 25 FN: 7
SMOTETomek	TP: 0 TN: 292 FP: 14 FN: 8	TP: 1 TN: 292 FP: 14 FN: 7	TP: 2 TN: 280 FP: 26 FN: 6	TP: 2 TN: 280 FP: 26 FN: 6

Figure 5.25: Results Predicting 15 Seconds After for Candidate D

Candidate E				
	Balanced Random Forest	Balanced Bagging Classifier	RUSBoost Classifier	Easy Ensemble Classifier
SMOTE	TP: 6 TN: 307 FP: 117 FN: 2	TP: 1 TN: 404 FP: 20 FN: 7	TP: 3 TN: 366 FP: 58 FN: 5	TP: 3 TN: 366 FP: 58 FN: 5
ADASYN	TP: 6 TN: 310 FP: 114 FN: 2	TP: 1 TN: 406 FP: 18 FN: 7	TP: 2 TN: 376 FP: 48 FN: 6	TP: 2 TN: 377 FP: 47 FN: 6
Borderline SMOTE	TP: 5 TN: 314 FP: 110 FN: 3	TP: 0 TN: 416 FP: 8 FN: 8	TP: 2 TN: 388 FP: 36 FN: 6	TP: 2 TN: 388 FP: 36 FN: 6
SVMSMOTE	TP: 0 TN: 411 FP: 13 FN: 8	TP: 2 TN: 414 FP: 10 FN: 6	TP: 1 TN: 389 FP: 35 FN: 7	TP: 2 TN: 393 FP: 31 FN: 6
SMOTEENN	TP: 8 TN: 184 FP: 240 FN: 0	TP: 2 TN: 382 FP: 42 FN: 6	TP: 4 TN: 365 FP: 59 FN: 4	TP: 3 TN: 363 FP: 61 FN: 5
SMOTETomek	TP: 0 TN: 405 FP: 19 FN: 8	TP: 1 TN: 405 FP: 19 FN: 7	TP: 3 TN: 366 FP: 58 FN: 5	TP: 3 TN: 366 FP: 58 FN: 5

Figure 5.26: Results Predicting 15 Seconds After for Candidate E

5.2.3 Third Experiment

With the results from the previous experiment, the main objective was to improve the outcomes from the best classifier and balancing technique, Balanced Random Forest Classifier and SMOTEENN. For this purpose, a round of tests performing a Grid Search with KFold and Nested Cross Validation was made. The pipeline consists of the balancing algorithm, the Robust Scaler, and the classifier. The scoring metrics observed were ROC AUC, Recall, Precision, F1 score, and Balanced Accuracy. The best algorithm was defined as the one with the best result from the F1 score. Therefore, the range of hyperparameters tested in this experiment follows:

Balanced Random Forest Classifier

- **Critetion**: [gini, entropy]
- **Max Depth**: [3, 4, 5]
- **N Estimators**: [100, 150, 200]
- **Min Samples Split**: [2, 3, 4, 5]
- **Min Samples Leaf**: [1, 2, 3, 4, 5]
- **Class Weight**: [balanced, balanced subsample]

- **Random State**: 1007

It is important to highlight that the best parameters for each candidate during the experiments did not reach their maximum points; this is, a max depth of 5, for example, in any hyperparameter. This way, we did not need to test with a wider range. The results are shown in Figures 5.28, 5.29 and 5.30. Moreover, 5.27 shows the results obtained during this experiment for the train data. The first, second and third number in each cell represents the result for the times of 5, 10 and 15 seconds, respectively.

Balanced Random Forest Classifier - SMOTEENN					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	86.2 | 77.4 | 77.5	70 | 53.3 | 85	10.7 | 8.4 | 8.3	18.4 | 14.3 | 15.1	76.5 | 67.4 | 77.8
B	84.3 | 66.9 | 69.6	76.7 | 33.3 | 73.3	9.1 | 3.9 | 4.5	16.1 | 6.8 | 8.5	78 | 55.5 | 68.1
C	84.9 | 80.6 | 78.7	60 | 55 | 58.3	12.1 | 8.7 | 8.8	19.9 | 15.1 | 14.8	74.5 | 70.8 | 71.4
D	70.5 | 78.1 | 76	55 | 65 | 50	10.4 | 9.8 | 6.7	17.3 | 17 | 11.8	70.6 | 74 | 66.9
E	86.9 | 70.4 | 70.2	63.3 | 43.3 | 61.7	11.3 | 4.4 | 4.2	18.8 | 7.9 | 7.9	76.8 | 63 | 68.1

Figure 5.27: Results Obtained from the Third Experiment with Balanced Random Forest Classifier.

Balanced Random Forest Classifier - SMOTEENN				
	TP	TN	FP	FN
A	6	116	147	2
B	7	193	100	1
C	4	214	94	3
D	8	233	86	0
E	8	342	85	0

Figure 5.28: Results Predicting 5 Seconds After.

Balanced Random Forest Classifier - SMOTEENN				
	TP	TN	FP	FN
A	8	146	116	0
B	6	166	126	2
C	5	230	76	3
D	6	213	94	2
E	7	120	306	1

Figure 5.29: Results Predicting 10 After.

Better predictions were obtained in this experiment when compared to the previous ones. Almost every case of falling was detected in the test sample, along with a lower rate of False Positives. The Balanced Random Forest Classifier, designed especially to deal with imbalanced data along with the SMOTEENN technique, an Under and Over sampling tool, shows better potential when compared to the other algorithms.

Balanced Random Forest Classifier - SMOTEENN				
	TP	TN	FP	FN
A	7	145	115	1
B	5	170	120	3
C	6	178	127	1
D	8	6	300	0
E	7	253	171	1

Figure 5.30: Results Predicting 15 Seconds After.

5.2.4 Fourth Experiment

We also implemented a Machine Learning architecture during the experimentation to test another category of algorithms to solve this problem. This way, we used a Perceptron architecture in the same test pipeline. Thus, we used the SMOTEENN algorithm to balance the data and a Robust Scaler for scaling the information along with the Perceptron algorithm. A Grid Search was performed in order not only to test this new technique but also to find its best parameters. After this, nested cross-validation was done. The range of hyperparameters for the Perceptron architecture follows:

Perceptron Architecture

- **Activation**: [identity, logistic, tanh, relu]
- **Solver**: [lbfgs, sgd, adam]
- **Learning Rate**: [constant, invscaling, adaptive]
- **Random State**: [1007]
- **Max Itter**: [300, 400, 500, 600]
- **Hidden Layer Sizes**: [(5, 5), (10, 10), (15, 15), (20, 20), (29, 29)]

It is important to explain the Hidden Layer Sizes hyperparameter; it represents the number of hidden layers and the number of neurons in each hidden layer. Therefore, *(5, 5)* represents a Perceptron architecture with two hidden layers and five neurons in each hidden layer. The range that represents the number of hidden layers and the number of neurons in each hidden layer was chosen based on (HEATON, 2008). Two hidden layers can represent an arbitrary decision boundary to arbitrary accuracy with rational activation functions and can approximate any smooth mapping to any accuracy. For the number of neurons, there are several rules to apply and guide this decision. The three rules followed were:

- The number of hidden neurons should be between the size of the input layer and the size of the output layer;
- The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer;
- The number of hidden neurons should be less than twice the size of the input layer.

Following these rules, we get a range to experiment. The results are shown in Figures 5.32, 5.33, 5.34. Moreover, 5.31 shows the results obtained during this experiment for the train data. The first, second and third number in each cell represents the result for the times of 5, 10 and 15 seconds, respectively. In addition, it is interesting to observe that despite the results obtained with the training data being good, the algorithm does not got reasonable results with the test data.

Perceptron					
	AUC	RECALL	PRECISION	F1	BALANCED ACCURACY
A	80.2 | 71 | 76.6	70 | 48.3 | 51.7	11 | 12.5 | 12.4	18.8 | 19.4 | 20	76.6 | 69.1 | 70.2
B	80.8 | 64.1 | 68	51.7 | 16.7 | 36.7	10.7 | 8.3 | 5.8	17.5 | 10.7 | 10	69.6 | 55.1 | 61.6
C	62.8 | 78.9 | 73.9	53.3 | 75 | 51.7	16.7 | 10 | 12.7	25.3 | 17.6 | 20.2	73.2 | 79.4 | 71.4
D	67.9 | 77.6 | 74.9	50 | 50 | 50	8.6 | 15.7 | 12.8	14.7 | 23.5 | 20.1	67.6 | 70.6 | 69.7
E	80.9 | 73.5 | 63.5	46.7 | 18.3 | 60	12 | 7.9 | 4.3	18.7 | 9.7 | 7.9	70.4 | 56.3 | 66.9

Figure 5.31: Results Obtained from the Fourth Experiment with Perceptron Classifier.

Perceptron Architecture				
	TP	TN	FP	FN
A	0	263	0	8
B	0	293	0	8
C	1	301	7	6
D	0	309	0	8
E	0	426	1	8

Figure 5.32: Results Predicting 5 Seconds After.

The results show that the Perceptron architecture did not get exciting results compared to all the previous tests. Yet, it is important to identify that the architecture

Perceptron Architecture				
	TP	TN	FP	FN
A	1	254	8	7
B	1	280	12	7
C	0	306	0	8
D	1	303	4	7
E	2	416	10	6

Figure 5.33: Results Predicting 10 Seconds After.

Perceptron Architecture				
	TP	TN	FP	FN
A	0	260	0	8
B	0	290	0	8
C	1	296	9	6
D	0	306	0	8
E	0	424	0	8

Figure 5.34: Results Predicting 15 Seconds After.

used in this experiment was one of the most simple in Machine Learning. Therefore, this might be an indication of the poor outcomes obtained.

6 Conclusion and Future Works

In this work, we presented an architecture to detect fall events in elderly people's homes. The tests and the proposal evaluation were based on an open-source dataset from a multi-agent architecture (KALUŽA et al., 2010). Although the outcomes obtained were not exciting enough to apply this solution in a true environment, they are a good starting point to investigate more about new techniques and tools. This is because numerous false alarms remain even with some good results in detecting fall events. In addition, the time interval between the event and the prediction still needs to be better. Thereby, some guidelines were followed here that can be changed in the future as the idea of having one specific model for each person, for example, in order to achieve better results.

Moreover, conducting new experiments with a Deep Learning architecture and use more tools to deal with imbalanced data might be interesting. This was another key point investigated in this work since it might occur when dealing with event prediction, depending on the source of the data.

Furthermore, some experiments should be explored more deeply regarding techniques and hyperparameter boosting. However, some variations were limited since the tests were carried out with a machine with low processing power. In addition, some preprocessing guidelines followed, if changed, might result in better outcomes for the algorithms and are going to be explored in the future as well.

In conclusion, event prediction, binary classification, imbalanced data, and Fog-Cloud architectures were the themes deeply studied during the construction of this work and provided good insights for the next steps in this field that will be explored to provide an architecture for event prediction.

Bibliography

ALIGHOLIAN, A. et al. Unsupervised event detection, clustering, and use case exposition in micro-pmu measurements. *IEEE Transactions on Smart Grid*, IEEE, v. 12, n. 4, p. 3624–3636, 2021.

AMARAL, W. D. do; DANTAS, M. A. Um modelo de reconhecimento de atividades humanas baseado no uso de acelerometro com qoc. *WSCAD-WIC 2017*, p. 45, 2017.

BOX, G. E. et al. *Time series analysis: forecasting and control*. [S.l.]: John Wiley & Sons, 2015.

BUUREN, S. V. *Flexible imputation of missing data*. [S.l.]: CRC press, 2018.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 41, n. 3, p. 1–58, 2009.

CHATFIELD, C. *The analysis of time series: an introduction*. [S.l.]: Chapman and hall/CRC, 2003.

DEGOEDE, K. M.; ASHTON-MILLER, J. A.; SCHULTZ, A. Fall-related upper body injuries in the older adult: a review of the biomechanical issues. *Journal of biomechanics*, Elsevier, v. 36, n. 7, p. 1043–1053, 2003.

DENG, R. et al. Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE internet of things journal*, IEEE, v. 3, n. 6, p. 1171–1181, 2016.

DUBOWSKY, S. et al. Pamm - a robotic aid to the elderly for mobility assistance and monitoring: a "helping-hand" for the elderly. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*. [S.l.: s.n.], 2000. v. 1, p. 570–576 vol.1.

EKLUND, J. et al. Information technology for assisted living at home: building a wireless infrastructure for assisted living. In: *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. [S.l.: s.n.], 2005. p. 3931–3934.

ESLING, P.; AGON, C. Time-series data mining. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 45, n. 1, p. 1–34, 2012.

GAMA, J. *Knowledge discovery from data streams*. [S.l.]: CRC Press, 2010.

GURALNIK, V.; SRIVASTAVA, J. Event detection from time series data. In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 1999. p. 33–42.

HEATON, J. *Introduction to neural networks with Java*. [S.l.]: Heaton Research, Inc., 2008.

HUANG, L. et al. Micro-seismic event detection and location in underground mines by using convolutional neural networks (cnn) and deep learning. *Tunnelling and Underground Space Technology*, Elsevier, v. 81, p. 265–276, 2018.

IDREES, S. M.; ALAM, M. A.; AGARWAL, P. A prediction approach for stock market volatility based on time series data. *IEEE Access*, IEEE, v. 7, p. 17287–17298, 2019.

KALUŽA, B. et al. An agent-based approach to care in independent living. In: SPRINGER. *International joint conference on ambient intelligence*. [S.l.], 2010. p. 177–186.

KASABOV, N. et al. Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke. *Neurocomputing*, Elsevier, v. 134, p. 269–279, 2014.

KHAN, D. U. et al. Designing a personal health application for older adults to manage medications. In: *Proceedings of the 1st ACM International Health Informatics Symposium*. New York, NY, USA: Association for Computing Machinery, 2010. (IHI '10), p. 849–858. ISBN 9781450300308. Disponível em: ⟨https://doi.org/10.1145/1882992.1883124⟩.

KHUAT, T. T.; LE, M. H. Ensemble learning for software fault prediction problem with imbalanced data. *International Journal of Electrical & Computer Engineering (2088-8708)*, v. 9, n. 4, 2019.

KIRAN, M. et al. Lambda architecture for cost-effective batch and speed big data processing. In: IEEE. *2015 IEEE International Conference on Big Data (Big Data)*. [S.l.], 2015. p. 2785–2792.

KRAWCZYK, B. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, Springer, v. 5, n. 4, p. 221–232, 2016.

KUMAR, V. et al. Addressing binary classification over class imbalanced clinical datasets using computationally intelligent techniques. In: MDPI. *Healthcare*. [S.l.], 2022. v. 10, n. 7, p. 1293.

MAKRIDAKIS, S.; WHEELWRIGHT, S. C.; HYNDMAN, R. J. *Forecasting methods and applications*. [S.l.]: John wiley & sons, 2008.

MAMUN, A. A. et al. A comprehensive review of the load forecasting techniques using single and hybrid predictive models. *IEEE Access*, v. 8, p. 134911–134939, 2020.

MARONNA, R. A. et al. *Robust statistics: theory and methods (with R)*. [S.l.]: John Wiley & Sons, 2019.

MARQUES, G. Ambient assisted living and internet of things. *Harnessing the Internet of everything (IoE) for accelerated innovation opportunities*, IGI Global, p. 100–115, 2019.

MARTÍNEZ-ÁLVAREZ, F. et al. A survey on data mining techniques applied to electricity-related time series forecasting. *Energies*, MDPI, v. 8, n. 11, p. 13162–13193, 2015.

MEHRMOLAEI, S.; KEYVANPOURR, M. R. A brief survey on event prediction methods in time series. In: *Artificial Intelligence Perspectives and Applications*. [S.l.]: Springer, 2015. p. 235–246.

MOUSTRA, M.; AVRAAMIDES, M.; CHRISTODOULOU, C. Artificial neural networks for earthquake prediction using time series magnitude data or seismic electric signals. *Expert systems with applications*, Elsevier, v. 38, n. 12, p. 15032–15039, 2011.

PATEL, A.; SHAH, J. Sensor-based activity recognition in the context of ambient assisted living systems: A review. *Journal of Ambient Intelligence and Smart Environments*, IOS Press, v. 11, n. 4, p. 301–322, 2019.

POLLACK, M. E. et al. Autominder: an intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems*, v. 44, n. 3, p. 273–282, 2003. ISSN 0921-8890. Best papers presented at IAS-7. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0921889003000770⟩.

QUDAH, I.; LEIJDEKKERS, P.; GAY, V. Using mobile phones to improve medication compliance and awareness for cardiac patients. In: *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*. New York, NY, USA: Association for Computing Machinery, 2010. (PETRA '10). ISBN 9781450300711. Disponível em: ⟨https://doi.org/10.1145/1839294.1839337⟩.

RADZUAN, N. F. M.; OTHMAN, Z.; BAKAR, A. A. Uncertain time series in weather prediction. *Procedia Technology*, Elsevier, v. 11, p. 557–564, 2013.

RASHIDI, P.; MIHAILIDIS, A. A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, IEEE, v. 17, n. 3, p. 579–590, 2012.

ROUET-LEDUC, B. et al. Machine learning predicts laboratory earthquakes. *Geophysical Research Letters*, Wiley Online Library, v. 44, n. 18, p. 9276–9282, 2017.

SADEGHI, M.; MCDONALD, A. D.; SASANGOHAR, F. Posttraumatic stress disorder hyperarousal event detection using smartwatch physiological and activity data. *Plos one*, Public Library of Science San Francisco, CA USA, v. 17, n. 5, p. e0267749, 2022.

SHUMWAY, R. H.; STOFFER, D. S.; STOFFER, D. S. *Time series analysis and its applications*. [S.l.]: Springer, 2000. v. 3.

SILVA, G. Di iorio et al. A watchdog proposal to a personal e-health approach. In: SPRINGER. *International Conference on Advanced Information Networking and Applications*. [S.l.], 2022. p. 81–94.

TEMKO, A. et al. Acoustic event detection and classification. In: *Computers in the human interaction loop*. [S.l.]: Springer, 2009. p. 61–73.

WANG, X. et al. Distributed arima models for ultra-long time series. *International Journal of Forecasting*, Elsevier, 2022.

ZHENG, X. *SMOTE variants for imbalanced binary classification: heart disease prediction*. [S.l.]: University of California, Los Angeles, 2020.