UNIVERSITÁ DEGLI STUDI DEL MOLISE
DOTTORATO IN BIOSCIENZE E TERRITORIO

MACHINE LEARNING AND FORMAL METHODS FOR SPORT ANALYTICS

DOCTORAL THESIS

[illegible]
Author
Umberto Antonio Di Giacomo

Tutor (s)
Prof. Antonella Santone Antonella Santone

Co-Tutor (s)
Prof. Giovanni Capobianco

The Coordinator of the PhD Program
Prof. Giovanni Fabbrocino

Pesche, March 2022

XXXIV

To my parents;
to my brothers Andrea, Antonio and Fabio;
to my sister.

Ringraziamenti

Finisce qui il mio percorso universitario che dura ormai da 8 anni. Ho iniziato l'università quasi per gioco, per passare il tempo e perchè non avevo altro da fare. Mi sono ritrovato a finire una triennale, una magistrale e un dottorato. Follia davvero! Quando ho iniziato non avevo idea di cosa aspettarmi da un ambiente come quello universitario. Spesso passavo le giornate in palestra a Pesche ma non ero mai andato nella parte più "interna", se non per prendere un tramezzino al bar: è lì che ho conosciuto Rosaria (mamma mia quanto parlava), per non dimenticare di Francesca e Stefano.

Ricordo il primo giorno che sono entrato "dentro" l'università. Credo che fosse fine Settembre, stavano per iniziare le lezioni (forse mancavano pochi giorni) e io chiaramente non avevo la minima intenzione di iniziare l'università. Il problema era che non avevo idea di cosa fare nella vita. Quella mattina mio padre mi ha quasi buttato giù dal letto, si può anche eliminare il quasi, costringendomi ad andare a parlare con un professore che mio padre e mia madre nominavano sempre a casa: il professor Capobianco. Ricordo perfettamente la scrivania del professore, per tutta la roba che c'era sopra non si vedeva neanche la scrivania e ricordo quasi ogni parola che il professore mi disse quel giorno. Chiaramente non me ne ero ancora reso conto, ma quel giorno probabilmente mi cambiò la vita. Ah, un altra cosa: dopo essere uscito dall'ufficio a mio padre venne l'idea di farmi conoscere un altro professore che aveva lo studio esattamente affianco a quello del professor Capobianco, il professor Fasano. Ci affacciammo nel suo studio e vidi questa persona che stava al computer (anzi, quasi dentro il computer perchè aveva gli occhi a 2 centimetri dallo schermo) e non appena ci vede disse: "Ah che piacere, tu sei il figlio di Michele. Sei pronto a programmare un po' e a usare Linux?". Chiaramente non avevo idea di cosa stesse dicendo.

Questo è stato il mio inizio. Purtroppo per voi che leggete, non posso parlare di tutto quello che ho passato all'università in 8 anni, dei pomeriggi passati in palestra con le mie squadre, delle giornate passate in aula studio fin quando non venivamo cacciati via da Egidio, addirittura di quella volta che abbiamo dormito all'università per l'Hackathon... momenti bellissimi. La verità è che l'Unimol per me è stata una seconda casa, una seconda famiglia, il posto dove ho passato forse i momenti più belli dei miei ultimi 8 anni, escludendo i campi di calcio chiaramente.

Non ho ancora, però, parlato della parte più bella di questi 8 anni: le persone. Sarà il luogo, sarà il Molise, sarà Isernia che rendono tutto più bello, ma le persone che ho incontrato all'Università sono state la cosa più bella.

Partiamo dai professori. Ho già nominato il professor Capobianco che per me è stato un riferimento incredibile, per tutto, non solo per aspetti accademici. Con lui ho potuto parlare di tutto, anche di calcio. Da lui ho imparato tantissimo. Ho nominato anche il professor Fasano, persona di una preparazione e di una simpatia uniche. Ma davvero tutti i professori che ho incontrato mi hanno lasciato qualcosa. Ringrazio anche il professor Mercaldo, che ho conosciuto soltanto negli ultimi 3 anni di dottorato e ringrazio tantissimo anche la professoressa Santone, che con me è stata sempre molto comprensiva nonostante alcune volte non rispettassi alcune scadenze che avevo, ma, purtroppo, il calcio veniva sempre prima di tutto.

Passiamo adesso alle persone più importanti che ho conosciuto in questo posto meraviglioso: gli amici.

Ringrazio Davide (l'Ingegnere) per l'immenso aiuto che mi hai dato ogni volta ti è stato chiesto. Mi sei stato vicino in tutti i momenti belli e brutti e in 8 anni ce ne sono stati di tutti i tipi. E pensare che all'inizio non mi eri neanche simpatico (forse non te l'ho mai detto Ing... ma tranquillo solo per il primo mese non mi eri molto simpatico).

Ringrazio Angelo, persona meravigliosa e di gran cuore. Grazie per tutte le risate che ci siamo fatti, per tutte le volte che mi hai cucinato a mezzanotte e, soprattutto, per avermi ospitato in quel periodo da te a Salerno, non lo dimenticherò mai.

Ringrazio Dania, per tutte le volte che mi hai fatto vedere le cose da un punto di vista diverso: praticamente ogni volta visto che non si trova mai una cosa su cui la pensiamo allo stesso modo, ma nonostante questo sai il bene che ti voglio.

Ringrazio Federica, Chiaretta, Luciana, Martina e Chiara per le tante serate e giornate passate insieme.

Ringrazio Simone (1berto, 2berti, 3berti), Andrew (per le tante camminate fatte insieme), Ilaria e Gerardo (di Ilaria ricordo soprattutto la sua eleganza mentre faceva lo skip alto in palestra), Paolo, Valentina e, per finire, Rosangela (soprattutto, per gli ultimi tre anni anche se dovresti ringraziarmi per la musica che ti ho fatto ascoltare nei viaggi da Isernia a Campobasso).

Ringrazio anche persone che non ho conosciuto all'università, ma che in questi anni ci sono sempre state e che per me sono fondamentali. Persone come Angioletto, Alessio, Pippo, Dani.

E per finire, ma sicuramente non perchè sono meno importanti (anzi!) ringrazio la mia famiglia, i miei genitori a cui devo tutto, mia sorella di cui sono molto fiero, i miei 4 nonni (anche i 2 che purtroppo oggi non ci possono essere), i miei zii... e Luna!

E alla fine ringrazio i miei 3 fratelli, la mia famiglia acquisita.

Ringrazio Antonio che c'è sempre stato da più di 20 anni ormai. Abbiamo avuto anche la fortuna di studiare nella stessa università... incredibile. Non c'è bisogno che dica nulla di più.

Ringrazio Fabietto, che mi vergogno quasi di considerare mio fratello minore perchè credo che sia anche più maturo di me. Ogni volta che ho bisogno di un punto di vista moderato su un argomento tu sei pronto a darmelo e sai quanto sei importante per me.

Ringrazio, per finire, Andrea. Lui sì che lo considero mio fratello minore. Probabilmente è la persona a cui voglio più bene al mondo, semplicemente per il suo modo di essere vero e instabile... un pò come me!

A questo punto spero di aver ringraziato veramente tutti. Sono tanti, lo so! Ma proprio per questo, proprio perchè sono tante le persone che ho vicino ogni giorno, devo essere grato. Sono grato di aver passato gli ultimi 8 anni della mia vita in un posto come l'Unimol. Auguro a tutti di trovare un luogo così. Auguro a tutti di incontrare le persone che ho incontrato io in questo percorso perchè persone così davvero che riescono a migliorarti la vita.

Grazie a tutti. Grazie Unimol!

Summary

The main focus of my PhD period is related to the concept of soccer analytics. During the last few years, soccer analytics has been growing rapidly. The main usage of this discipline is the prediction of soccer match results, even if it can be applied with interesting results in different areas, such as analysis based on the player position information. This context has been explored through three different steps: the analysis and recognition of human activities, the adoption of machine learning on soccer data and, at the end, the application of formal methods on sport analytics scenario. In this way, I want to explore the strengths and the weaknesses of different techniques.

Human activity recognition is attracting interest from researchers and developers in recent years due to its immense applications in wide area of human endeavors. The main issue in human behaviour modeling is represented by the diverse nature of human activities and the scenarios in which they are performed. These factors make this aspects challenging to deal with.

Then, machine learning techniques have been used in order to make some consideration on a great amount of data related to soccer matches. Specifically, these type of techniques have been used in order to predict soccer game results and player positions during a match.

The last step is about the usage of formal methods in order to provide more explainability and interpretability of the results obtained. With the application of formal methods based approach, I try to detect the playing style of a soccer teams, providing transparency of the results obtained. I model soccer teams in terms of automata and, by exploiting model verification techniques, I verify whether a playing style, expressed by means of a temporal logic formula, is exhibited by the team under analysis. This information can support the coach in determining the strategy of the team while the match is in progress. The experimental analysis confirms the effectiveness of the proposed method in soccer team behaviour detection, obtaining promising results, compared with standard baseline approaches.

List of publications

International Journals

1. Giovanni Capobianco, Umberto Di Giacomo, Francesco Mercaldo, Antonella Santone. (2021). Machine Learning on Soccer Players Position *International Journal of Decision Support System Technology (IJDSST).*

International Conferences/Workshops with Peer Review

1. G. Capobianco, U. Di Giacomo, F. Mercaldo, A. Santone (2018, December, 10 – 13). A Methodology for Notational Analysis and Real-Time Decision Support in Sport Environment. *Proceedings of IEEE BigData, Seattle, WA, USA.*

2. G. Capobianco, U. Di Giacomo, F. Mercaldo, V. Nardone, A. Santone (2019, February 19-21). Can Machine Learning Predict Soccer Match Results? *Proceedings of 11th International Conference on Agents and Artificial Intelligence, ICAART, Prague – Czech Republic.*

3. G. Capobianco, U. Di Giacomo, F. Mercaldo, V. Nardone, A. Santone (2019, February 13-15). Wearable Devices for Human Activity Recognition and User Detection. *Proceeding of 27th Euromicro International Parallel, Distributed, and Network-Based, PDP, Pavia, Italy.*

4. G. Capobianco, U. Di Giacomo, F. Mercaldo, A. Santone (2019). Dunuen: A User-Friendly Formal Verification Tool. *23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Procedia Computer Science, Budapest.*

5. Giovanni Capobianco, Umberto Di Giacomo, Francesco Mercaldo, Antonella Santone, and Tommaso Di Tusa (2019). A Methodology for Real-Time Data Verification exploiting Deep Learning and Model Checking. *Proceedings of IEEE BigData.*

6. R.Casolare, U. Di Giacomo, F. Martinelli, F. Mercaldo, A. Santone (2020). Android Collusion Detection by means of Audio Signal Analysis with Machine Learn-

ing techniques. *25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*.

7. U. Di Giacomo, R. Casolare, O. Eigner, F. Martinelli, F. Mercaldo, T. Priebe, A. Santone (2020). Exploiting Supervised Machine Learning for Driver Detection in a Real-World Environment. *25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems.*.

Contents

1
1.1 Background 1
1.2 Objective of the study 4

5
2.1 Behavioral Analysis 5
2.2 State of the art 6
2.3 Human Activity Recognition 9
2.3.1 The dataset 10
2.3.2 Classification analysis 12
2.3.3 Human Activity Recognition Outcomes 13
2.3.4 User Detection Outcomes 14
2.4 Driver Detection 16
2.4.1 The Dataset 18
2.4.2 Descriptive statistics 19
2.4.3 Classification analysis 21

25
3.1 State of the art 25
3.2 Soccer Match Outcome 27
3.2.1 Method 28
3.2.2 Dataset 30
3.2.3 Evaluation 31
3.3 Soccer Player Position 35
3.3.1 Method 35
3.3.2 Dataset 38
3.3.3 Evaluation 39

49
4.1 State of the art 49
4.2 Dunuen 50

4.2.1 Technique and Implementation 50
4.2.2 Overall Architecture 52
4.2.3 Case Study 54
4.3 Real-Time Data Verification 55
4.3.1 The Method 56
4.4 Style Detection 59
4.4.1 The Method 60
4.4.2 Results 62
4.4.3 Discussion and Comparison 69

77
5.1 Conclusion 77

79

CHAPTER 1

Introduction

1.1 Background

Sport analytics consists in the investigation and modelling of professional sports performances and contests using scientific techniques, as machine learning methods. This discipline frequently employs principles and techniques from statistics, data mining, game theory, bio-mechanics, kinesiology with the aim to take informed decisions and gain a competitive sport advantage. In other words, sport analytics is the practice of applying mathematical and statistical principles to different sports, such as baseball [30], basketball [47] and hockey [58]. Although each sport has its own characteristics, sport analytics uses the same basic methods and approaches as any other kind of data analysis and, when properly applied, can yield tremendous competitive advantages to a team or an individual player. In soccer, the most usage is about the prediction of results and the definition of strategies that can be used to win a game or to obtain an improvement of the team performances. Usually, the models constructed in these analysis are based on several aspects about the game, such as tactical, technical or physical information. In my work, machine learning and formal methods techniques have been used and I am going to present them more in detail.

Machine learning is a type of artificial intelligence able to provide computers with the ability to learn without being explicitly programmed [65]. The tasks are typically classified into two categories, depending on the nature of the learning available to a specific system [64]:

- *Supervised learning*: the computer is presented with example inputs and their desired outputs, given by a “teacher", and the goal is to learn a general rule that maps inputs to outputs. It represents the classification: the process of building a model of classes from a set of records that contains class labels.

- *Unsupervised learning*: no labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

The methods proposed in this thesis consider supervised learning: these algorithms are the most widespread to solve data mining problems [65] such as, for example, malware detection problem [15], pathology classification [63] or driver style recognition problem [62].

The supervised learning approach is composed of two different steps:

1. **Construction of a model**: starting from the data that we have available, we construct the model that it will be used, in the next step, for the classification of the remains data that are not used in this step.
2. **Prediction**: the model constructed in the previous step is used in order to classify the new data.

For the evaluation of the classification analysis, the following metrics have been used: Precision, Recall, F-Measure and ROC Area.

The precision has been computed as the proportion of the examples that truly belong to class X among all those which were assigned to the class. It is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved:

$$Precision = \frac{tp}{tp+fp}$$

where *tp* indicates the number of true positives and *fp* indicates the number of false positives.

The recall has been computed as the proportion of examples that were assigned to class X, among all the examples that truly belong to the class, i.e., how much part of the class was captured. It is the ratio of the number of relevant records retrieved to the total number of relevant records:

$$Recall = \frac{tp}{tp+fn}$$

where *tp* indicates the number of true positives and *fn* indicates the number of false negatives.

The F-Measure is a measure of a test's accuracy. This score can be interpreted as a weighted average of the precision and recall:

$$F\text{-}Measure = 2 * \frac{Precision*Recall}{Precision+Recall}$$

The Roc Area is defined as the probability that a positive instance randomly chosen is classified above a negative randomly chosen.

Now, let's consider the second type of method. Formal methods are mathematical techniques, often supported by tools, for developing software and hardware systems. Mathematical formalism allows the users to analyze and verify models during the program life-cycle: requirements engineering [37, 82], specification, architecture [96, 97], design [96, 97], implementation, testing, maintenance, and evolution [26].

Formal methods can be considered as "the applied mathematics for modeling and analyzing ICT systems" [7]. Their aim is to establish system correctness with mathematical formalism. Their great potential has led to an increasing use by engineers of formal methods for the verification of complex software and hardware systems. Also, formal methods are one of the highly recommended verification techniques for software development of safety-critical systems [14, 27] according to several practices standard, such as International Electrotechnical Commission (IEC) and European Space Agency (ESA). During the last two decades, research in formal methods has led to the development of some very promising verification techniques that facilitate the early detection of defects [25, 61].

These techniques are accompanied by powerful software tools that can be used to automate various verification steps [41, 42]. Formal methods rely on different formal verification techniques such as model based verification algorithm, e.g. model checking.

Model-based verification techniques are based on models that describe all the possible system states in a mathematically precise and rigorous manner. The system models are accompanied by algorithms that systematically explore all the possible states. This provides the basis for a whole range of verification techniques ranging from an exhaustive exploration (model checking) to experiments with a restrictive set of scenarios [13, 53, 68]. Due to unremitting improvements of underlying algorithms and data structures, together with the availability of faster computers and larger computer memories, model-based techniques that a decade ago only worked for very simple examples are nowadays applicable to realistic designs.

Model checking is one of the most widespread verification techniques. It explores all possible system states in a brute-force manner. Similar to a computer chess program that checks possible moves, a model checker, the software tool that performs the model checking, examines all possible system scenarios in a systematic manner. In this way, it can be shown that a given system model truly satisfies a certain property.

The general model checking workflow comprises the following steps:

- **Model construction:** Creating an abstract representation of the system;
- **Property specification:** Encoding the formal specification describing the desired/expected system behaviour;
- **Verification:** Automatically verifying the correctness of the model related to the formal specification.

There are many model checkers developed in the world. Several tools in last years were developed for formal verification. From an historical perspective, CADP and SPIN seem to be the two oldest model checkers still available. For example, *Construction and Analysis of Distributed Processes* (CADP) [36] is a comprehensive software toolbox that implements the results of concurrency theory. Another model checker is NUSMV [24] which has been developed by Carnegie Mellon University (CMU) and Istituto per la Ricerca Scientifica e Tecnologica (IRST). This model checker is designed to be a well structured, open, flexible and documented platform for model checking. NUSMV is the result of the reengineering and reimplementation of the CMU SMV symbolic model checker. PRISM [55], instead, is a probabilistic model checking tool being developed at the University of Birmingham. In the case of probabilistic model checking, the model are probabilistic, in the sense that they encode the probability of making a transition between states. PRISM [55] considers probabilistic models i.e., it encodes the probability of making a transition between states. One factor that limits the use of model checking (and of the formal verification tools we cited) to specialists is related to the deep knowledge of formal specification languages, i.e., to provide a support tool that automatically accomplish the Model construction.

1.2 Objective of the study

The goal of my research is to understand if is it possible to model the behavior of human beings in different contexts, starting from a real-world environment context (driver detection and activity recognition with accelerometer data) to a soccer-based context.

The analysis performed is based on three different steps:

- *Step 1*: The first step is about the study of behavioral analysis that aims to recognize activities from a series of observations on the actions of subjects and the environmental conditions. The application of this context include video surveillance, health care, and human-computer interaction.
- *Step 2*: In the second step, it has been applied machine learning techniques in order to predict the result of a soccer match, starting from data of different nature (technical, tactical and physical), and to predict the position of the players during a match.
- *Step 3*: In the final step, formal methods are used trying to recognize the team's style of play and, at the same time, obtaining a more explainability and interpretability of the results obtained.

All these steps are analyzed more in detail in the next chapters. In the Chapter 2, I analyze the Behavioral Analysis context more in details; in the Chapter 3, I apply the machine learning method in the context of Soccer Analytics; in the Chapter 4 it has been used Formal methods on Soccer Analytics; in the Chapter 5 I analyze the conclusion of my work.

CHAPTER 2

Behavioral Analysis

2.1 Behavioral Analysis

In recent years, behavioural analysis recognition is attracting interest from researchers and developers due to its immense applications in wide area of human endeavors. The main issue in human behaviour modeling is represented by the diverse nature of human activities and the context in which they are performed.

In this chapter it has been proposed two different studies performed in different contexts: Human Activity Recognition (HAR) and driver detection. Both studies are based on human activity data.

The first one is aimed to recognize human activities and detect users, performing these activities, using features gathered from accelerometer sensors widespread in wearable and mobile devices. I exploit machine learning techniques to build models with the ability to discriminate between a set of user activities: sitting, sitting down, standing, standing up and walking. Furthermore, it has been demonstrated that the proposed method is able to distinguish between different users and identify the user genre.

The second study is about driver identification. Driver identification consists of discovering the driver identity of a running vehicle based on what he possesses and/or on his physical and behavioral characteristics [44]. This topic is recently becoming very critical for the automobile industry, achieved by an increasing number of more and more sophisticated and accurate car sensors and monitoring systems able to extract information about the driver (e.g., hand geometry, keystroke dynamics, voice-print). The main motivation of driver identification is the awareness that it may improve the driver's experience allowing a safer and more comfortable driving, an intelligent assistance in case of emergencies and even a reduction of global environmental problems [73]. Looking as an example at the driver security, driver identification may allow discovering which member of the family is currently driving and consequently perform an automatic setting of the car equipment (e.g., radio volume and frequency, temperature) [95]. Furthermore, the driver identification may support to detect changes in the driver behavior (due to possible indisposition or state of being drunk) and activate according security procedures (for example, a notification asking the driver to stop soon). Finally, driver identification can be useful to suggest new car improvements based on the driver preferences or new systems to reduce the gas consumption and pollution based on the driving characteristics. The study of the driver behavior with respect to each segment of a road may also allow to profile each road section supporting the activation of alert signals when more caution is required (for example a vocal message may alert the driver to reduce the brake pressure in a dangerous curve) [46]. Based on the above discussed advantages, deriving from the the driver profiling and identification, several studies have been proposed in the last years focusing on the identification of driver physical and behavioral features. The firsts [29, 40] are stable human characteristics that have been largely diffused in the banking and forensic domains to guarantee an higher safeness with respect to the more traditional authentication system based on the ownership of a key (this authentication system can be easily by-passed when someone gets a hold of the key). The seconds consist to detect individual personality features which is becoming object of several studies in recent years that are mainly focused on speaker recognition [78]. The limit of these approaches is that they are based on the analysis of only one behavioral feature. This may cause an high uncertainty in driver identification specially if there is a noisy sensor. For this reason new approaches based on multi-modal identification systems are introduced [35, 79]. They provide a more accurate driver identification basing on the detection and analysis of an higher set of behavioral features.

2.2 State of the art

In this section, I review the current state-of-art with regard to user detection, human activity recognition and driver detection. Firstly, let's consider the state of the art about user detection and human activity recognition. eWatch [8] is an activity recognition framework which embeds sensors and a micro-controller within a device that can be worn as a sport watch. The considered sensors to detect the human activities are: accelerometer, a light sensor, a thermometer, and a microphone. In order to discriminate between different activities, authors build models using decision tree and time-domain algorithm, obtaining an overall accuracy equal to 92.5% for six ambulation activities, although they achieved less than 70% for activities such as descending and ascending.

Vigilante [56] is an Android application for human activity recognition. The Zephyr's BioHarness BT [3] chest sensor strap was used to measure acceleration and physiological signals: heart rate, respiration rate, breath waveform amplitude, and skin temperature, among others. The decision tree classifier is able to detect three ambulation activities with an overall accuracy of 92.6%. The application can run for up to 12.5 continuous hours with a response time of no more than 8% of the window length. Different users with diverse characteristics participated in training and testing phases, ensuring flexibility to support new users without the need to re-train the system.

The method proposed in [90] is designed to detect ambulation and gymnasium activities such as lifting weights, rowing, doing push up with different intensities (a total of 30 activities). In the experiment they consider 21 participants, consider subject-dependent and subject-independent from the study. They consider machine learning to train models from the gathered features obtaining an accuracy equal to 94.6% for subject-dependent analysis, while 56% of accuracy was obtained in the subject-independent evaluation. The proposed method gathers features through five accelerometers placed on the user's dominant arm and wrist, hip, thigh, and ankle, as well as a heart rate monitor on the chest. Finally, the integration device is a laptop, which allows for better processing capabilities, but prevents portability and pervasiveness.

ActiServ is an activity recognition service for mobile phones [10, 11]. Authors consider a fuzzy inference system aimed to classify ambulation and phone activities based on the signals given by the built-in accelerometer only. From the performance point of view, when ActiServ is executed to meet a real-time response time, the accuracy drops to 71%, while ActiServ can also reach up to 90% after personalization, in other words, a subject-dependent analysis. Authors reported confusion matrices of the classification, showing that the activity labeled as walking was often confused with cycling, while standing and sitting could not be differentiated when the mobile device orientation was changed.

Riboni et al. [80] presented a method for context-aware activity recognition exploiting statistical and ontological reasoning under the Android platform. The proposed system is able to recognize human activities as well as brushing teeth, strolling, and writing on a blackboard. They gather data from two accelerometers, one in the mobile device and another on the user wrist. Authors also discuss the statistical classification of activities considering the historical variant. For instance, whether the predictions for the last five time windows were {jogging, jogging, walking, jogging, jogging}, the third window was likely a misclassification (i.e., due to the user performing some atypical movement) and the algorithm should automatically correct it. The obtained accuracy was roughly 93%.

Authors in [50] present an activity detection method based on an accelerometer sensor placed on the user's dominant wrist. They apply time domain features and the Linear Discriminant Analysis in order to reduce the dimension of the feature space. Then, a Fuzzy Basis Function learner considering fuzzy If-Then rules, is used to classify the several activities. An accuracy equal to 94.71% was reached for seven activities: brushing teeth, hitting, knocking, working at a PC, running, walking, and swinging.

Ugolino and colleagues [91] consider 5 different activity classes, gathered from 4 subjects wearing accelerometers mounted on their waist, left thigh, right arm, and right ankle. Authors consider decision tree machine learning-based algorithm used in connection with the AdaBoost ensemble method for classifying different activities. The overall recognition performance was of 99.4% (weighted average) using a 10-fold cross validation testing mode.

Authors in [22] propose a technique for user's authentication and verification using gait as a biometric pattern. They consider the accelerometer sensors to extract features from the Android mobile device of users under analysis. They built a dataset from 22 person using the accelerometer sensor in the upper torso of a person, obtaining an accuracy ranging from 87.1 to 94.26.

Voigt et al. [94] explore the possibility of using point cloud data gathered from wearable depth cameras for on-body activity recognition. They consider 16 participants performing nine distinct activities (i.e., cleaning, cooking, dishes, eating, book, sleeping, phone, TV and PV) in three home environments. They build supervised machine learning models using 10-fold cross-validation with the KNN and Random Forest classification algorithms, obtaining an F-Score ranging 0.53 (eating) to 1 (sleeping activity).

Researchers in [72] considere as feature set the accelerometer and gyroscopes placed at the ankle, chest, hip and wrist with 19 participants performing eleven activities (i.e., sitting, lying, standing, washing dishes, vacuuming, sweeping, walking, ascending stair, descending stair running and jumping). In the first experiment, the performances of these sensors were analyzed using seven machine-learning classification algorithms. In the second experiment, they evaluated the feature gathered from sensors attached on the same location and on different locations of the body. Considering 10-fold cross validation they demonstrated that the fusion of heterogeneous sensors attached to different locations of the body shows Chest and Hip sensors fusion achieves an average F-measure of 0.942 and classification accuracy of 94.23% using Random Forest algorithm.

Let's consider now the state of the art about driver detection.

Considering the growing trend of cyber-attacks targeting vehicles, security technology has been studied and developed. As results of these efforts, the ISO 26262[1], an international standard for functional safety of electrical and/or electronic systems in production automobiles defined by the International Organization for Standardization (ISO) in 2011, was published. ISO 26262 is intended to be applied to safety-related systems that include one or more electrical and/or electronic (E/E) systems and that are installed in series production passenger cars with a maximum gross vehicle mass up to 3500 kg, while does not address unique E/E systems in special purpose vehicles such as vehicles designed for drivers with disabilities.

In order to develop and establish an open industry standard for automotive E/E architecture, the AUTOSAR (AUTomotive Open System ARchitecture)[2][3] development partnership was formed in July 2003 by BMW, Bosch, Continental, DaimlerChrysler, Siemens VDO and Volkswagen. In November 2003, Ford Motor Company joined as a Core Partner. It pursues the objective of creating and establishing an open and standardized software architecture for automotive electronic control units (ECUs) excluding infotainment. Goals include the scalability to different vehicle and platform variants, transferability of software, the consideration of availability and safety requirements, a collaboration between various partners, sustainable utilization of natural resources, maintainability throughout the whole "Product Life Cycle". In the following, I review the current literature related to the driving style recognition. A hidden-Markov-model-(HMM)-based similarity measure is proposed in [33] in order to model driver human behavior. They employ a simulated driving environment to test the effectiveness of the proposed solution.

The authors of [66] consider cepstral features of each driver obtained through spectral analysis of driving signals are modeled with a GMM: GMM driver model based on cepstral features is evaluated in driver identification experiments using driving signals

[1]http://www.iso.org/iso/catalogue_detail?csnumber=43464
[2]https://www.elektrobit.com/products/ecu/technologies/autosar/
[3]http://www.autosar.org/

collected in a driving simulator and in a real vehicle on a city road. They categorized the observed driving signals into three groups: Driving behavioral signals (e.g., gas pedal pressure, brake pedal pressure, and steering angle); Vehicle status signals (e.g., velocity, acceleration, and engine speed) and Vehicle position signals (e.g., following distance, relative lane position, and yaw angle). Experimental results show that the driver model based on cepstral features achieves a driver identification rate of 89.6% for driving simulator and 76.8% for real vehicle, resulting in 61% and 55% error reduction, respectively, over a conventional driver model that uses raw driving signals without spectral analysis. The authors of [89] propose an algorithm for real-time driver identification by using the combination of unsupervised anomaly detection and neural networks. Their algorithm extracted nonphysiological signals as input, namely, driving behavior signals from inertial sensors (e.g., accelerometers) and geolocation signals from GPS sensors. Their approach is able to identify the drivers within 13 seconds and 81% accuracy, regardless of routes and traffic conditions. The study of [92] focuses on a lightweight, end-to-end deep-learning framework for performing driver-behavior identification. The proposed architecture features depth-wise convolution, along with augmented recurrent neural networks for time-series classification. They also demonstrate the robustness in order to show that the accuracy of driver identification algorithms is not directly associated with the reliability of car sensors, which are prone to malfunction, noise, failures, and hacking attempts. The approach compare the performance of several algorithms for driver identification with an accuracy rate between 95% and 98%. Nor and colleagues [71] use the kernel density estimation (KDE) as tools to extract features. Then, they adopt these features to recognize the emotion of the driver by using Multilayer Perceptron (MLP) as classifiers. The data collection was conducted in Singapore during vacation time. The driver must have at least two years driving experience. They managed to collect 11 drivers including men and women, aged between 24-25 years old. The considered features are the brake and gas pedal pressures. In the experiment they state that each driver meets the accuracy level which is more than 50%: the highest accuracy is obtained from driver 10 with an accuracy equal to 71.94%, while the lowest accuracy is obtained from driver 3 with an accuracy equal to 61.65% in classification according to the other driver. Naturalistic data from University of Texas Drive (UTDrive) corpus have been used by Choi et al. [23] to derive both GMM and HMM models for the sequence of driving characteristics (wheel angle, brake pedal status, acceleration status, and vehicle speed). The authors have shown that driver identification can be accomplished at rates ranging from 30 to 70%. Another good result in drivers' recognition was performed by Zhang et al. [99] who used HMM to analyze the data of the accelerator and steering wheel of each driver, and achieved an accuracy of 85%.

2.3 Human Activity Recognition

In this section, it has been explained in detail the proposed method about user detection and human activity recognition. Figure 2.1 depicts the proposed approach.

The analysis made is based on the following research questions:

- RQ1: is it possible to recognize human activities using information gathered from the accelerometer sensor?

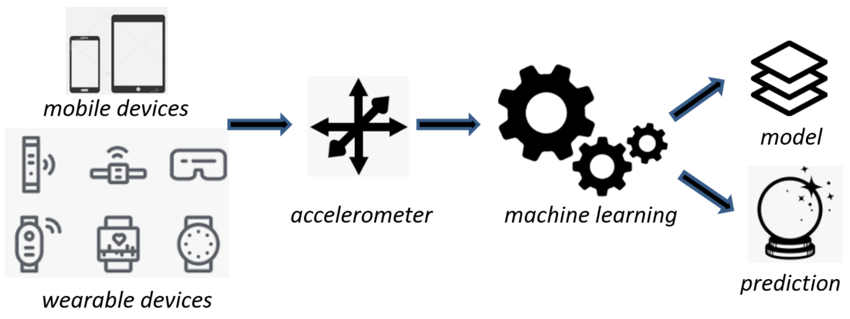


Figure 2.1: *The adopted approach: from wearable and mobile devices I obtain features from accelerometer sensor. These features are the input for machine learning tasks (i.e., model and prediction).*

- RQ2: is it possible to discriminate between users using information gathered from the accelerometer sensor?

As stated into the first section, one of the distinctive characteristics of the proposed method is the usage of a common feature set based on the accelerometer sensor for user and human activity detection. Accelerometers are cheap sensors and they are available in wearable and mobile devices. An accelerometer is able to measure proper acceleration. Proper acceleration, being the acceleration (or rate of change of velocity) of a body in its own instantaneous rest frame, it is not the same as coordinate acceleration, being the acceleration in a fixed coordinate system.

Accelerometers are electromechanical devices that sense either static or dynamic forces of acceleration. Static forces include gravity, while dynamic forces can include vibrations and movements.

Accelerometers can measure acceleration on one, two, or three axes. 3-axis units (i.e., x, y and z) are becoming really common in wearable and mobile devices as the cost of development for them decreases.

Generally, accelerometers contain capacitive plates internally. Some of these are fixed, while others are attached to minuscule springs that move internally as acceleration forces act upon the sensor. As these plates move in relation to each other, the capacitance between them changes. From these changes in capacitance, the acceleration can be determined.

As depicted in Figure 2.1 the x, y and z featured gathered from accelerometer sensors represent the input for the machine learning task.

Now, different aspects of the analysis have been presented. Specifically, the considered dataset used to evaluate the proposed solution in a real-world scenario; the adopted procedures to build and test the models (i.e., classification analysis) and the analysis performed (Human Activity Recognition and the User Detection).

2.3.1 The dataset

The study is based on two different datasets composed by real-world features gathered from accelerometers: the first one (i.e., D1 dataset) is obtained using wearable devices [2], while the second one (i.e., D2 dataset) is obtained using a mobile device [1]. Both the D1 dataset and the D2 dataset are publicly available for research purposes.

Table 2.1: *Users involved D1 in dataset with details about genre, age, height and weight.*

User	Genre	Age	Height	Weight
#1	woman	46	1,62 m	67 Kg
#2	woman	28	1,58 m	53 Kg
#3	man	31	1,71 m	83 Kg
#4	man	75	1,67 m	67 Kg

The wearable devices used to build the #D1 dataset are comprised of 4 tri-axial ADXL335 accelerometers connected to an ATmega328V microcontroller. All modules were of the Lilypad Arduino toolkit. The positioning of accelerometers is shown in Figure 2.2.

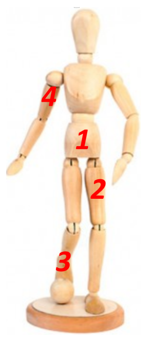


Figure 2.2: *Wearable devices scheme of positioning: (1) waist, (2) left thigh, (3) right ankle and (4) right arm.*

The four accelerometers were respectively positioned in the waist (1), left thigh (2), right ankle (3), and right arm (4). All accelerometers were calibrated prior to the data collection. The calibration consists of positioning the sensors and the performance of the reading of values to be considered as “zero”. From the calibration, the read values of each axis during data collection are subtracted from the values obtained at the time of the calibration.

The collected data are related to 8 hours of activities, 2 hours with each one of the 4 users. Table 2.1 shows details about the users involved in the D1 dataset.

The D2 dataset is related to data extracted from an Android smartphone positioned in the chest pocket from 22 participants walking in the wild over a predefined path.

Differently from the D1 dataset, in this case there is just one accelerometer (i.e., the one provided by the mobile device).

Acceleration data from walking activities have been acquired using a Google Nexus One mobile phone with operating system Android 2.2. Android-based mobile phones have an open Application Program Interface that allows programming the phone and accessing the sensors present in it to read and save accelerometer data. Each sensor in the Android platform has an listener associated that delivers data when a change in its value happen. The delivery rate can be set to different frequency but this value is just an hint to the system. Accelerometer has been sampled with a timestamp of approximately 33 ms, using the mobile phone in normal mode, with all the network connections active.

Data have been collected from 22 users with ages between 25 and 35. Users perform seven different runs of walking, for a total of 140 different walking runs. Users were free to perform all the runs as they like. The mobile phone has been put in the jacket pocket on the chest. The acceleration axis are concordant to the specification of the Android platform and, in our setting, the z axis refers to the direction concordant to the movement. The walking activity is performed in indoor, outdoor and urban environment.

2.3.2 Classification analysis

The classification analysis consists in building classifiers in order to evaluate the feature vector accuracy to distinguish between (i) human activities (*sitting*, *sitting down*, *standing*, *standing up* and *walking*), (ii) different users and (iii) genre (*woman* and *man*). This is the reason why I train four different classifiers: the first one related to human activity recognition (with D1 dataset), the second one related to user detection (with D1 dataset), the third one related to user detection (with D2 dataset), while the last one related to genre identification (with D1 dataset).

For training the classifiers (represented by the machine learning model task in Figure 2.1),I defined T as a set of labeled feature trace *(M, l)*, where each M is associated to a label $l \in$ *{sitting, sitting down, standing, standing up, walking}* with regard to the human activities recognition. Relating to user detection, each M is associated to a label $l \in$ *{user #1, user #2, user #3, user #4, user #5, user #6, user #7, user #8, user #9, user #10, user #11, user #12, user #13, user #14, user #15, user #16, user #17, user #18, user #19, user #20, user #21, user #22}* while, for the genre identification task each M is associated to a label$l \in$ *{woman, man}*.

For each M I built a feature vector F $\in R_y$, where y is the number of the features used in training phase ($y = 12$) when I consider the D1 dataset (x, y and z axis for the four accelerometers of wearable devices).

For each M I built a feature vector F $\in R_y$, where y is the number of the features used in training phase ($y = 3$) when I consider the D2 dataset (x, y and z axis for the accelerometer of mobile devices).

For the learning phase,I use a k-fold cross-validation [5]: the dataset is randomly partitioned into k subsets. A single subset is retained as the validation dataset in order to evaluate the obtained model, while the remaining $k-1$ subsets of the original dataset are considered as training data.I repeated the process for $k = 2$ times; each one of the k subsets has been used once as the validation dataset [69, 87]. To obtain a single estimate, I computed the average of the k results from the folds.

I evaluated the effectiveness of the classification method with the following procedure:

1. build a training set $T \subset D$;
2. build a testing set $T' = D \div$T;
3. run the training phase on T;
4. apply the learned classifier to each element of T'.

In the approach depicted in Figure 2.1 the evaluation of the classifier is represented by the machine learning prediction task.

Table 2.2: *Precision, Recall, F-Measure and Roc Area for Human Activity Recognition.*

Alg.	Precision	Recall	F-Measure	Roc Area	Activity
	0.999	0.999	0.999	0.999	sitting
J48	0.933	0.924	0.928	0.980	sitting down
	0.989	0.988	0.989	0.995	standing
	0.928	0.921	0.925	0.972	standing up
	0.976	0.981	0.979	0.991	walking
	0.982	0.983	0.983	0.984	sitting
BN	0.899	0.889	0.924	0.928	sitting down
	0.979	0.974	0.975	0.981	standing
	0.888	0.893	0.889	0.969	standing up
	0.956	0.969	0.969	0.978	walking
	0.993	0.991	0.989	0.989	sitting
SVM	0.921	0.898	0.926	0.932	sitting down
	0.982	0.983	0.983	0.985	standing
	0.885	0.888	0.901	0.976	standing up
	0.962	0.973	0.972	0.981	walking
	0.995	0.992	0.992	0.992	sitting
DT	0.944	0.922	0.908	0.934	sitting down
	0.983	0.986	0.984	0.984	standing
	0.888	0.889	0.907	0.981	standing up
	0.968	0.975	0.974	0.983	walking

Each classification was performed using 50% of the dataset as training dataset and 50% as testing dataset employing the full feature set.

In order to enforce the conclusion validity, in this study I consider four classification algorithm: J48, BayesNet (BN), LibSVM (SVN) and DecisionTable (DT) [65].

2.3.3 Human Activity Recognition Outcomes

In this section, the results for the Human Activity Recognition have been presented

The experiment is performed using the D1 dataset and the results are shown in Table 2.2.

For all the considered activities the recognition rate is very high. The algorithm obtaining the best performances from the point of view of the analysed metrics is J48: it obtains for the sitting activity a precision and a recall equal to 0.999, while for the standing activity a precision equal to 0.898 and a recall equal to 0.988. Furthermore, also the walking activity recognition reaches good values for precision (0.976) and recall (0.981). Slightly lower performances are obtained with regard to standing up (0.928 of precision and 0.921 of recall) and sitting down activities (0.933 of precision and 0.924 of recall). With regard to the remaining algorithms, they obtain slightly lower performances but in general the average of precision and recall is almost equal to 0.9.

RQ1 response: The information gathered from accelerometer sensors embedded into four wearable devices are promising to distinguish between different human activities with a precision ranging between 0,928 (standing up activity) and 0.999 (sitting activity), while the recall is ranging between 0.921 (standing up activity) and and 0.999 (sitting activity) using the J48 algorithm.

Table 2.3: *Precision, Recall, F-Measure and Roc Area for User detection evaluated for D1 dataset.*

Alg.	Precision	Recall	F-Measure	Roc Area	Class
	0.983	0.986	0.984	0.993	user #1
J48	0.980	0.976	0.978	0.990	user #2
	0.979	0.981	0.980	0.991	user #3
	0.977	0.972	0.975	0.992	user #4
	0.978	0.979	0.981	0.987	user #1
BN	0.977	0.974	0.975	0.986	user #2
	0.978	0.977	0.976	0.979	user #3
	0.971	0.968	0.969	0.988	user #4
	0.979	0.983	0.981	0.988	user #1
SVM	0.979	0.974	0.974	0.977	user #2
	0.977	0.981	0.978	0.987	user #3
	0.975	0.971	0.973	0.986	user #4
	0.981	0.982	0.978	0.989	user #1
DT	0.978	0.977	0.977	0.984	user #2
	0.978	0.978	0.979	0.982	user #3
	0.974	0.973	0.972	0.984	user #4

2.3.4 User Detection Outcomes

Below are presented the results obtained with regard to User Detection. The results of following experiments have been presented :

1. user detection using D1 dataset;
2. user detection using D2 dataset;
3. genre identification using D1 dataset.

In Table 2.3 are shown the results obtained for user detection using D1 dataset (note that the D1 dataset is composed from features obtained from accelerometers of four wearable devices).

Even in this experiment, the best performances are obtained from the J48 algorithm: precision and recall is ranging between 0.977 and 0.983, while recall is ranging from 0.972 to 0.986. The performances obtained from the remaining algorithms are almost equal to 0.978 in terms of precision and recall.

Tables 2.4, 2.5, 2.6 and 2.7 show the results for user detection using the D2 dataset (i.e., the one considering only one accelerometer sensor embedded in the user mobile device) with the J48, the BayesNet, the LibSVM and the DecisionTable algorithms.

Considering the J48 algorithm, the precision is ranging from 0.941 to 0.958 while recall is ranging from 0.942 to 0.966: results are slightly lower if compared with the ones obtained in the user detection with the D1 dataset (Table 2.3): this is symptomatic that even if with one accelerometer the proposed method is able to obtain, in the worst case, a precision equal to 0.941 and a recall equal to 0.942 (user #18), increasing the number of accelerometer is it possible to slightly increase the performances: the worst case of the experiment with the D1 dataset reports a precision equal to 0.977 and a recall equal to 0.972 (user #3 in Table 2.4). With regard to the BayesNet (Table 2.5) the best precision obtained is equal to 0.949 (users #2 and 21), while the best recall is equal to 0.958 (users #1, #2, #4, #5 and #22). Almost equal performances are obtained

Table 2.4: *Precision, Recall, F-Measure and Roc Area for User detection evaluated for D2 dataset with the J48 algorithm.*

Precision	Recall	F-Measure	Roc Area	Class
0.958	0.967	0.953	0.757	user #1
0.951	0.961	0.952	0.758	user #2
0.949	0.959	0.951	0.751	user #3
0.953	0.966	0.953	0.761	user #4
0.955	0.966	0.955	0.756	user #5
0.948	0.949	0.946	0.751	user #6
0.949	0.953	0.948	0.756	user #7
0.946	0.956	0.952	0.759	user #8
0.950	0.959	0.958	0.763	user #9
0.952	0.962	0.959	0.765	user #10
0.953	0.963	0.962	0.766	user #11
0.948	0.955	0.956	0.762	user #12
0.952	0.962	0.962	0.765	user #13
0.945	0.948	0.947	0.755	user #14
0.946	0.946	0.948	0.756	user #15
0.947	0.948	0.949	0.754	user #16
0.943	0.949	0.946	0.755	user #17
0.941	0.942	0.946	0.749	user #18
0.943	0.945	0.949	0.752	user #19
0.949	0.954	0.951	0.768	user #20
0.952	0.963	0.961	0.766	user #21
0.954	0.963	0.962	0.768	user #22

from the LibSVM and DecisionTable algorithms: as a matter of fact, with regard to the first one the best precision (0.933) is obtained for the user #13 and the best recall (0.943) is obtained for the user #4. Slightly lower performances are reached from the DecisionTable algorithm: it obtains as best precision a value equal to 0.892 (user #14) and a best recall equal to 0.897 (users #12).

Table 2.8 shows the results obtained for the genre identification using the D1 dataset.

Considering the J48 algorithm, the woman genre is identified with a precision and a recall equal to 0.99, while the man genre is identified with a precision equal to 0.984 and a recall equal to 0.983: this is symptomatic that from the point of view of the considered features (the accelerometers embedded in the wearable devices) men and women are distinguishable because the two categories exhibit different values for the analysed features. In order to confirm the performances related to the other experiments, even in this case the J48 outperforms the other classification algorithms: as a matter of fact,a precision equal to 0.982 (for the woman class) and equal to 0.977 (for the man one) is obtained using the LibSVM algorithm. The BayesNet obtain a precision equal to 0.987 (for the woman class) and 0.985 (for the man one), while the DecisionTable algorithm reaches a precision equal to 0,984 (for the woman class) and 0.982 (for the man one).

A similar trend is showed with regard to recall: using the LibSVM algorithm is equal to 0.981 (for the woman class) and 0.979 (for the man one), while with the BayesNet algorithm is equal to 0.987 (for the woman class) and to 0.985 (for the man one). The DecisionTable classification reaches a recall equal to 0.984 (with regard to the woman class) and 0.982 (with regard to the man one).

Table 2.5: *Precision, Recall, F-Measure and Roc Area for User detection evaluated for D2 dataset with the BayesNet algorithm.*

Precision	Recall	F-Measure	Roc Area	Class
0.946	0.958	0.946	0.739	user #1
0.949	0.958	0.947	0.741	user #2
0.943	0.951	0.946	0.743	user #3
0.946	0.958	0.948	0.752	user #4
0.947	0.958	0.949	0.748	user #5
0.931	0.941	0.944	0.748	user #6
0.942	0.949	0.943	0.747	user #7
0.939	0.946	0.946	0.747	user #8
0.939	0.949	0.949	0.751	user #9
0.947	0.953	0.951	0.754	user #10
0.947	0.956	0.957	0.752	user #11
0.942	0.951	0.949	0.753	user #12
0.944	0.955	0.955	0.756	user #13
0.941	0.941	0.941	0.749	user #14
0.938	0.937	0.939	0.745	user #15
0.943	0.940	0.941	0.743	user #16
0.939	0.938	0.937	0.744	user #17
0.933	0.936	0.938	0.738	user #18
0.938	0.936	0.941	0.742	user #19
0.941	0.948	0.948	0.755	user #20
0.946	0.956	0.950	0.753	user #21
0.949	0.958	0.956	0.755	user #22

RQ2 response: the information gathered from accelerometer sensors exhibits values able to discriminate between different users. The experiment with the D1 dataset (built with 4 wearable devices) showed that the precision is ranging from 0.977 and 0.983, while the recall from 0.972 and 0.986. The experiment with the D2 dataset (built considering only the accelerometer embedded the user smartphone) showed that with one accelerometer the performances are slightly lower, however they exhibit good performances. The last experiment, the one aimed to identify the genre, demonstrated that feature extracted from wearable devices can be also considered to distinguish between women and men with high performances. Best performances are obtained using the J48 classification algorithm.

2.4 Driver Detection

In this section it has been described the method used to identify driver behavior, using data retrieved from the CAN bus. In order to collect the data, On Board Diagnostics 2 (OBD-II) and CarbigsP as OBD-II scanners are used. Every recent vehicle has many measurement sensors and control sensors and is managed by a Electronic Control Unit (ECU). ECU is the device that controls parts of the vehicle such as engine, automatic transmission, and antilock braking system (ABS). OBD refers to the self-diagnostic and reports capability by monitoring vehicle system in terms of ECU measurements and vehicle failures. The data is recorded every 1 second during driving. It has been considered a real-world environment and not a simulated one in order to examine all possibly relevant variables, for instance: slowdowns at traffic lights and other possible

Table 2.6: *Precision, Recall, F-Measure and Roc Area for User detection evaluated for D2 dataset with the LibSVM algorithm.*

Precision	Recall	F-Measure	Roc Area	Class
0.924	0.941	0.932	0.716	user #1
0.928	0.942	0.934	0.723	user #2
0.926	0.941	0.933	0.721	user #3
0.931	0.943	0.935	0.727	user #4
0.933	0.940	0.935	0.721	user #5
0.926	0.939	0.934	0.719	user #6
0.938	0.938	0.936	0.721	user #7
0.929	0.937	0.935	0.722	user #8
0.925	0.938	0.937	0.722	user #9
0.927	0.939	0.939	0.724	user #10
0.926	0.939	0.939	0.725	user #11
0.929	0.939	0.938	0.723	user #12
0.931	0.938	0.936	0.722	user #13
0.929	0.939	0.934	0.725	user #14
0.928	0.936	0.933	0.725	user #15
0.929	0.932	0.935	0.724	user #14
0.928	0.936	0.932	0.723	user #17
0.929	0.931	0.934	0.726	user #18
0.928	0.931	0.931	0.727	user #19
0.931	0.933	0.932	0.726	user #20
0.928	0.937	0.931	0.728	user #21
0.929	0.934	0.933	0.731	user #22

variables that are not considerable in a simulated environment. More specifically, the experiment is aimed at verifying whether the feature set is able to discriminate the car owner from impostors.

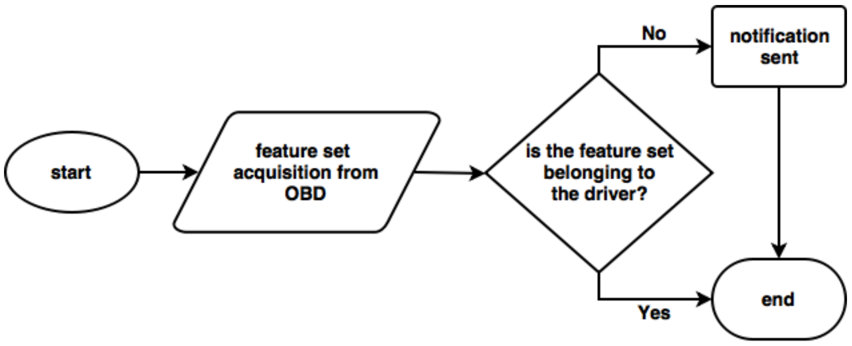

start
feature set acquisition from OBD
is the feature set belonging to the driver?
No
notification sent
Yes
end

Figure 2.3: *Flow diagram of the proposed approach.*

Figure 2.3 represents a flow diagram of the proposed approach. The most important step is represented by the selection block: in this block the feature set is acquired from the OBD and it is evaluated against the learned model in order to decide whether it belongs to the car owner.

The classification analysis was accomplished with Weka[4], a suite of machine learning software, largely employed in data mining for scientific research.

[4]`http://www.cs.waikato.ac.nz/ml/weka/`

Table 2.7: *Precision, Recall, F-Measure and Roc Area for User detection evaluated for D2 dataset with the DecisionTable algorithm.*

Precision	Recall	F-Measure	Roc Area	Class
0.889	0.895	0.878	0.701	user #1
0.893	0.896	0.876	0.704	user #2
0.884	0.895	0.873	0.710	user #3
0.889	0.891	0.878	0.714	user #4
0.881	0.892	0.878	0.713	user #5
0.882	0.894	0.876	0.715	user #6
0.885	0.893	0.875	0.709	user #7
0.883	0.894	0.876	0.712	user #8
0.886	0.896	0.879	0.715	user #9
0.885	0.894	0.881	0.714	user #10
0.884	0.893	0.880	0.709	user #11
0.886	0.897	0.879	0.709	user #12
0.886	0.889	0.875	0.711	user #13
0.892	0.898	0.873	0.718	user #14
0.889	0.895	0.876	0.709	user #15
0.888	0.894	0.882	0.709	user #14
0.879	0.891	0.879	0.707	user #17
0.887	0.890	0.882	0.712	user #18
0.888	0.891	0.882	0.713	user #18
0.892	0.893	0.892	0.716	user #20
0.884	0.889	0.884	0.709	user #21
0.891	0.892	0.891	0.719	user #22

On the next section, it has been presented the real-world dataset used in the analysis and the results of the experiment.

For the sake of clarity, the results of the evaluation will be discussed reflecting the data analysis' division in three phases: descriptive statistics, hypotheses testing and classification.

2.4.1 The Dataset

Ten different drivers participated to the experiment by driving, with the same car, 4 different round-trip path in Seoul for about 23 hours of total driving time. It has been used only one car for the experimental analysis, since I want that the variables used are affected only by the driver's characteristics and not by external factors, like for example the path or the type of car used.

The driving path consists of three types of city way, motor way and parking space with the total length of about 46 km. The experiments was performed in the similar time zone from 8 p.m. to 11 p.m. on weekdays. The ten drivers completed two round trips for reliable classification, while data are collected from totally different road conditions. The city way has signal lamps and crosswalks, but the motor way has none. The parking space is required to drive slowly and cautiously.

The data that I used has total 94,401 records stored every second with the size of 16.7Mb in total and it is freely available for research purpose[5].

[5] `https://sites.google.com/a/hksecurity.net/ocslab/Datasets/driving-dataset`

Table 2.8: *Precision, Recall, F-Measure, Roc Area for User detection evaluated for genre identification.*

Alg.	Precision	Recall	F-Measure	Roc Area	Genre
J48	0.990	0.990	0.990	0.995	woman
	0.984	0.983	0.984	0.995	man
SVM	0.982	0.981	0.979	0.981	woman
	0.977	0.979	0.977	0.977	man
BN	0.987	0.987	0.988	0.987	woman
	0.985	0.986	0.978	0.982	man
DT	0.984	0.985	0.985	0.985	woman
	0.982	0.983	0.973	0.976	man

2.4.2 Descriptive statistics

Figure 2.4 shows the box plots for the 10 different drivers involved in the study related to the *Accelerator_pedal_value* defined as the degree to which the driver is depressing the accelerator pedal. This feature ranges between 0% and 100%. Considering the big number of features, I do not show the box plot related to the full set composed of the 51 features, but a similar consideration can be done for all the features involved in the study.

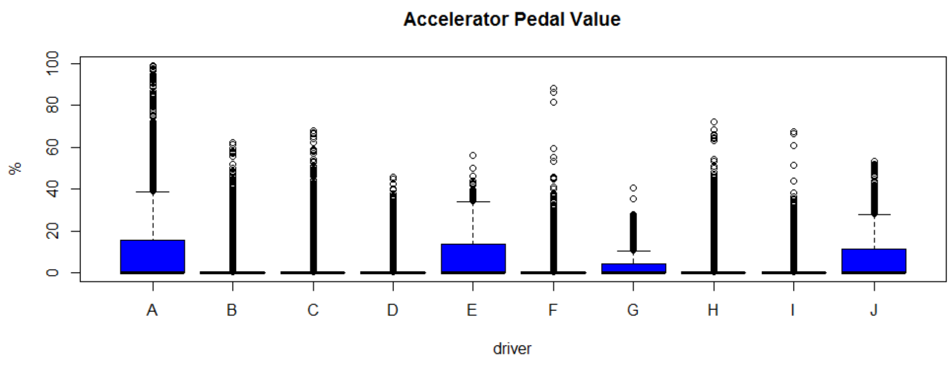


Figure 2.4: *Box plots related to the Accelerator_pedal_value feature for the ten different drivers considered in the experiment.*

The box plots show that the considered feature exhibits a similar distribution for A and E drivers (with a maximum acceleration degree equal to 20%), while for B, C, D, F, H and I drivers the acceleration degree is closed to 0%. Drivers G and J present a medium acceleration degree if compared with the previous drivers groups. Our idea is that A and E drivers carry more pressure on accelerator pedal with respect to other drivers and, consequently, B, C, D, F, H and I drivers performing a lower pressure on the accelerator pedal, they can be considered prudent drivers. E and J drivers, considering their medium accelerator pedal pressure, can be considered the average drivers in this analysis.

Figure 2.5 shows the box plots for the 10 different drivers involved in the study related to the *Torque_of_friction* defined as torque caused by the frictional force that occurs. This feature ranges between 0% and 100%. The friction is the mechanism that allows drivers to take advantage of the potential of the vehicle, without it is not possible to make hill starts or change gear quickly. It allows the separation between the drive shaft and gear thus allowing to carry out a change of gear and with its slip and transmit a torque capacity allows us to move the vehicle in both the flat and uphill. When the friction pedal is pressed, through a mechanical or hydraulic system, the friction generates a pressure of plate mechanism and the respective thrust bearing, and the disc of the high coefficient of friction is removed making the free torque transmission, and consequently, the delivered engine power. The friction pedal must be pressed exclusively at the start, stop and during gear changes. Many people use it more than necessary, even on the march, this causes premature damages of the clutch.

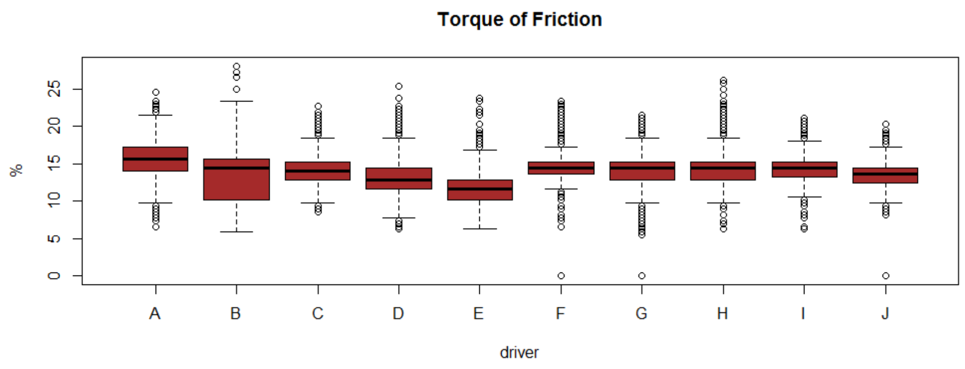


Figure 2.5: *Box plots related to the Torque_of_friction feature for the ten different drivers considered in the experiment.*

The box plots show that all drivers involved in the experiment present similar distributions. As a matter of fact, only the B driver presses the friction pedal for a longer time than the others, while F, G, H, I and J drivers exhibit very similar distributions. For instance, correlating the *Accelerator_pedal_value* and the *Torque_of_friction* features,I can state the driver B is a very prudent driver, but need to press for less time the friction pedal in order to prevent corrupting, while driver A even he/she guides faster (and therefore presumably must change gear more often), it spends less time in pressing the friction pedal.

The box plots in Figure 2.6 present the distributions for the 10 drivers related to the *Fuel_consumption* feature. This feature ranges between 0 and 10000 and it is measured in cubic millimeter (mcc).

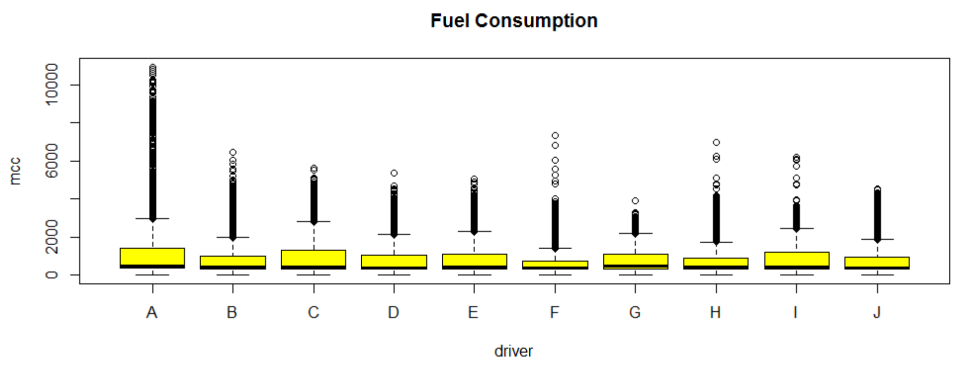


Figure 2.6: *Box plots related to the the Fuel_consumption feature for the ten different drivers considered in the experiment.*

The 10 drivers present a similar distribution; considering that the *Fuel_consumption* feature can be correlated to the *Accelerator_pedal_value* (more pressure on the accelerator pedal increases the speed of the vehicle and then more fuel is required) the driver A presents a slightly larger box plot if compared with other drivers. The box plots in Figure 2.7 show the driver distribution related to *Intake_air_pressure* feature (i.e., the pressure of air inhaled to engine). It ranges between 0 and 255, and it is measured in Kilopascal (kPA).

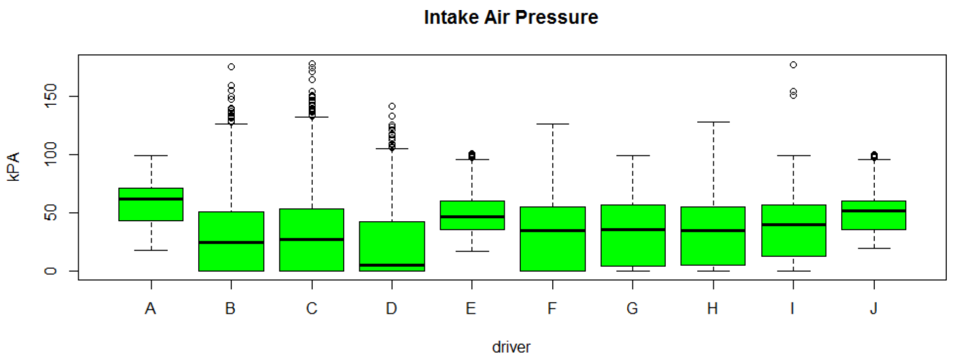


Figure 2.7: *Box plots related to the Intake_air_pressure feature for the ten different drivers considered in the experiment.*

From the analysis of the *Intake_air_pressure*, it seems that engines of A, E and J drivers inhale similar pressure of air (ranging between 45 and 60 kPA) while, the remaining engine drivers (B, C, D, F, G, H and I) need to an air pressure ranging between 0 and 50 kPA. Considering that the air intake pressure is influenced by the degree of opening throttle plate which draws the fuel to the combustion chamber in engine [4], a bigger air pressure will be reflected in fuel increased consumption.

This is consistent about our analysis about A driver and, considering the *Accelerator_pedal_value* feature I highlight that bigger pressure on pedal accelerator is performed also by A, E and J drivers: the same drivers who engines require more air pressure.

2.4.3 Classification analysis

The results obtained are shown in Table 2.9. In particular, the table shows the FP Rate, Precision, Recall, F-Measure and Roc Area for classifying the full drivers dataset with the multi-driver classification. The time column is related to the time (in seconds) to learn the classifier. In the all drivers classification the time to learn the algorithms is ranging between 4.4s (with RepTree algorithm) and 16.18s (with J48consolidated algorithm).

Two feature selection algorithms employed, the BestFirst and the GreedyStepwise, confirm that 6 features on the 51 considered in the full features dataset are the most discriminatory in driver identification i.e., the #5 (*Intake_air_pressure*), the #8 (*Engine_soak_time*), the #12 (*Long_Term_Fuel_Trim_Bank1*), the #15 (*Torque_of_friction*), the #35 (*Transmission_oil_temperature*) and the #50 (*Steering_wheel_speed*) features.

Table 2.9 shows the classification analysis considering the features retrieved from the feature selection step.

Table 2.9: *Best Features Classification results.*

Family	Algorithm	FP Rate	Precision	Recall	F-Measure	Roc Area	Time
All drivers	J48	0.001	0.989	0.989	0.989	0.998	1.69s
	J48graft	0.001	0.990	0.99	0.990	0.997	1.58s
	J48consolidated	0.002	0.986	0.986	0.986	0.998	1.71s
	Random Tree	0.002	0.984	0.984	0.984	0.991	0.41s
	RepTree	0.002	0.984	0.984	0.984	0.998	0.54s
Driver A	J48	0.000	0.998	0.997	0.998	0.999	1.33s
	J48graft	0.000	0.997	0.997	0.997	0.999	1.29s
	J48consolidated	0.000	0.997	0.997	0.997	0.999	1.36s
	Random Tree	0.000	0.997	0.996	0.997	0.998	0.25s
	RepTree	0.000	0.997	0.996	0.997	1.000	0.49s
Driver B	J48	0.001	0.991	0.991	0.991	0.998	3.26s
	J48graft	0.002	0.990	0.992	0.991	0.998	3.18s
	J48consolidated	0.002	0.988	0.985	0.987	0.998	3.23s
	Random Tree	0.002	0.986	0.987	0.986	0.992	0.35s
	RepTree	0.002	0.986	0.987	0.986	0.998	0.47s
Driver C	J48	0.002	0.980	0.979	0.980	0.997	2.36s
	J48graft	0.002	0.982	0.979	0.98	0.996	2.25s
	J48consolidated	0.002	0.972	0.983	0.977	0.997	2.45s
	Random Tree	0.002	0.974	0.971	0.972	0.984	0.39s
	RepTree	0.002	0.973	0.972	0.972	0.997	0.42s
Driver D	J48	0.002	0.987	0.984	0.985	0.997	4.71s
	J48graft	0.002	0.987	0.986	0.986	0.996	4.79s
	J48consolidated	0.002	0.985	0.973	0.979	0.997	4.69s
	Random Tree	0.003	0.980	0.980	0.980	0.988	0.46s
	RepTree	0.003	0.980	0.974	0.977	0.997	0.79s
Driver E	J48	0.000	0.997	0.997	0.997	0.999	1.64s
	J48graft	0.000	0.998	0.997	0.998	0.999	1.58s
	J48consolidated	0.000	0.996	0.996	0.996	0.999	1.62s
	Random Tree	0.001	0.995	0.996	0.995	0.998	0.27s
	RepTree	0.000	0.996	0.995	0.995	0.999	0.49s
Driver F	J48	0.001	0.990	0.991	0.990	0.998	1.23s
	J48graft	0.001	0.990	0.991	0.991	0.998	1.27s
	J48consolidated	0.002	0.984	0.987	0.986	0.998	1.25s
	Random Tree	0.002	0.986	0.988	0.987	0.993	0.32s
	RepTree	0.002	0.984	0.986	0.985	0.999	0.31s
Driver G	J48	0.000	0.995	0.995	0.995	0.999	2.28s
	J48graft	0.001	0.994	0.995	0.994	0.999	2.34s
	J48consolidated	0.001	0.991	0.994	0.992	0.998	2.31s
	Random Tree	0.001	0.989	0.989	0.989	0.994	0.38s
	RepTree	0.001	0.989	0.990	0.989	0.999	0.51s
Driver H	J48	0.002	0.986	0.988	0.987	0.998	4.11s
	J48graft	0.001	0.988	0.987	0.988	0.997	4.07s
	J48consolidated	0.002	0.982	0.986	0.984	0.998	4.05s
	Random Tree	0.002	0.980	0.981	0.980	0.989	0.53s
	RepTree	0.003	0.978	0.982	0.980	0.998	0.85s
Driver I	J48	0.002	0.982	0.985	0.983	0.996	2.79s
	J48graft	0.001	0.984	0.985	0.985	0.997	2.67s
	J48consolidated	0.002	0.976	0.984	0.980	0.996	2.87s
	Random Tree	0.002	0.979	0.974	0.977	0.986	0.28s
	RepTree	0.002	0.976	0.982	0.979	0.998	0.43s
Driver J	J48	0.001	0.988	0.986	0.987	0.997	2.51s
	J48graft	0.001	0.987	0.987	0.987	0.997	2.43s
	J48consolidated	0.001	0.986	0.983	0.984	0.997	2.56s
	Random Tree	0.003	0.976	0.978	0.977	0.988	0.33s
	RepTree	0.002	0.981	0.976	0.979	0.997	0.51s

In the classification the time to learn the algorithms is ranging between 0.41s (with the Random Tree algorithm) and 1.71s (with the J48consolidated algorithm). Without PCA analysis the J48consolidated algorithm employs 16.18s to learn the classifier. The obtained results in terms of the analyzed metrics are closed to the previous ones, confirming that the excluded features were not useful in the classification task. Indeed, in the second classification it has been used only 6 features on the 51 considered in the previous one: the use of 6 features instead of 51 is reflected in a higher applicability of the proposed method in the real world. As a matter of fact, the use of a lower number of features is reflected in a smaller storage space and in a shorter computing time in order to identify the driver impostor.

Using the best features (All drivers family), these are the results:

- FP rate equal to 0.001 with the J48 and J48graft algorithms;
- Precision, Recall and F-Measure equal to 0.989 using the J48 and the J48graft classification algorithms;
- Roc Area equal to 0.998 using J48, J48consolidated, and RepTree classification algorithms.

In order to have a full vision about the single driver identification, I represent using histogram the obtained performance by the classification algorithms only for the FP rate considering the classification algorithm able to reach the best performance in the best feature classification i.e., the J48 algorithm.

Figure 2.8 shows the Precision, obtained classifying using the six best features with the J48 algorithm, exhibit by the 10 drivers involved in the experiment.

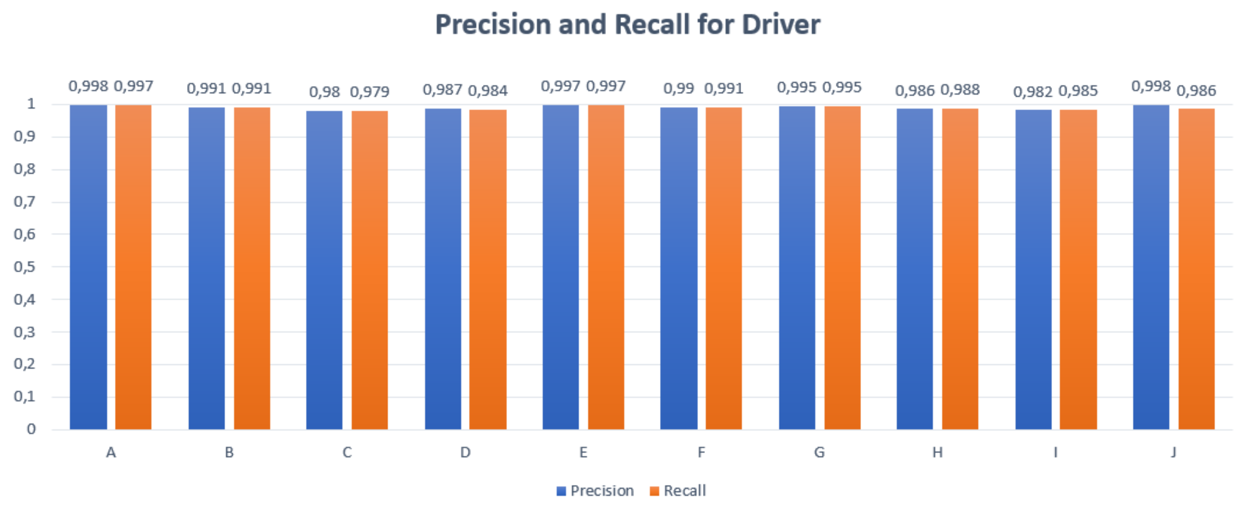


Figure 2.8: *Precision and Recall values for the 10 drivers involved in the experiment obtained classifying the six best features using the J48 algorithm.*

The Precision ranges between 0.98 and 0.998 for all the drivers involved in the experiment. This result demonstrates that our method is able to identify on 100 retrieved instances 98 or 99 that are belonging to the right class (i.e., owner or impostor). In detail, drivers A, B, E, F, G, and J obtain a precision equal or greater than 0.99; while drivers C, D, H and I reach a precision equal or greater than 0.98.

Figure 2.8 shows also the Recall, obtained classifying using the six best features with the J48 algorithm, exhibit by the 10 drivers involved in the experiment.

The Recall ranges between 0.979 and 0.997 for all the drivers involved in the experiment, i.e., is the fraction of relevant instances that are retrieved. The best recall value is obtained by drivers A (0.997), E (0.997), G (0.995), B (0.991) and F (0.991). The remaining drivers reach a recall value greater than 0.98 with the only exception of C driver with a recall equal to 0.979.

CHAPTER 3

Machine Learning on Soccer Analytics

3.1 State of the art

The analysis that can be performed with sport analytics is typically divided into two different parts: bio-mechanical and notational analysis [43]. Both techniques involve the analysis and improvement of the sport performance giving good feedbacks to coaches and athletes. Sport biomechanics is concerned with fine detail about individual sport techniques while, on the other hand, notational analysis is more concerned with gross movements or movement patterns in games or teams, and is primarily interested in strategy and tactics. These types of analysis are useful because, if I consider the Ill-chosen performance indicators to evaluate a specific game, it can be possible to highlight advantageous strategy or important aspect of team performance. In other words, they help coaches to identify good and bad performances of team member or the whole team [9].

Typically, the main weakness of the current state-of-the-art research is two-fold. The first one relates to methodologies while the second one relates to the evaluation of the proposed methods.

With regard to the methodology weakness, literature presents methods analyzing feature set available only at the end of matches, for instance the number of goal or the number of red cards received by the players.

This is the reason why it is not possible to predict the result of a match in progress, and this represents a limitation because in this way the coach is not able to change the tactic for instance, between the first and the second time.

The second weakness is related to the evaluation of the proposed method. Probably for the novelty of the topic, the researchers do not have available a dataset of real-data to analyse the proposed solution and to compare its effectiveness with the other methods. This is an important issue, because currently it is not possible to compare the performances of the various methodologies proposed. This problem is discussed by

Rein and colleagues [77]: they stated that one of the main problems in sport analytics is the lack of available and relevant data and this is becoming an obstacle in itself for the modelling of tactical decision making in team sports.

From the other side, big data researchers affirm that, with the development of advanced tracking technologies, this situation will change in a while. Indeed, they sustain that the amount of available data related to sport analytics is becoming increasingly difficult to manage [59, 88]: I will assist to the opposite situation, researchers will have a lot of data available and the difficult task will be the extraction of knowledge from heterogeneous sources.

One of the most widespread use of sport analytics, in soccer environment, is related to the prediction of soccer match results. In literature there are several works focused about the most important factors that influence the results of a game. Clearly, the winning of a match is not related just to one factor [60], but there are several aspects that contribute to this end. Considering how widespread are sports, there is an increasing interest in developing methodologies and techniques aimed to predict a match result examining a set of indicators [31] (generally based on statistics related to previous matches).

In [19], the authors propose a new feature set aimed to model a soccer match. The set is related to characteristics obtainable not only at the end of the match, and it is used to predict the results of the match and the number of goals scored by the team that won the game.

In [54], the author has evaluated a set of machine learning algorithms in order to classify the 3 label *result* variables i.e., *win, draw* and *loss*: authors observed the best performance with the Multilayer Perceptron algorithm with an accuracy equal to 69.474 % and a ROC area equal to 0.836.

In [49], an approach based on Bayesian Networks to predict match results has been presented. The analysis shold that the Bayesian networks is generally superior to other techniques such as the MC4, a decision tree learner, naive Bayesian learner (NB), and k-nearest neighbor learner (KNN) for this domain in terms of predictive accuracy. Specifically, authors obtain an accuracy equal to 59% which outperformed other machine learning models i.e., 41.7% (obtained by the MC4 classification algorithm), 47.86% (with the NB algorithm) and 50.58% (with the KNN algorithm). A similar analysis has been proposed in [57] it has been considered the problem of predicting the outcome of a soccer match finished with a draw at the end of the first half using mainly the information stored during the first part of the match. The dataset considered in this study contains the results of 166 matches and, after removing the attributes containing few values different from zero, the final set of feature is composed of 27 attributes. Firstly, the authors have used a feature selection method and, then a classification task on a three label *result* variables - *home_win*, *away_win* or *draw*. The RBF network is the algorithm that performs better, with an accuracy equal to 45%.

In [85] the authors have used multiple classification algorithms such as support vector machine, random forest and Naive Bayes. The best performing algorithm in that study is SVM, having an accuracy equal to 0.599 followed by Naive Bayes with an accuracy equal to of 0.55. Random forest is the algorithm with worst performance, with an accuracy equal to 0.50.

Many studies have been performed on Premier League matches. In [76], it has been used a Bayesian networks approach. This research uses predictors as home and away team shots, home and away team shots on target, home and away corners, half time home and away team goals, and so on, to predict the match outcome of a team. In particular, the study considered the English Premier League for the seasons of 2010-2011, 2011-2012 and 2012-2013. It has been used a 10-folds cross-validation to perform the classification, and the results have an accuracy, in average,equal to 74%. In that research, the authors use predictors that give direct information on the outcome of a specific match, so variables that are strongly correlated to the predicted variable, as goals scored in the first and second half.

In [6] the authors demonstrate the building of a generalized predictive model for predicting the results of the English Premier League. Firstly, the authors have used feature engineering and exploratory data analysis to create a feature set for determining the most important factors for predicting the results of a football match. Then, a highly accurate predictive system using machine learning has been created. The best model of that study uses gradient boosting, achieving a performance of 0.2156 on the ranked probability scores (RPS) metric for game weeks 6 to 38 for the English Premier League. A different feature set is considered in [39]: they obtain a lower average accuracy if compared to the method I propose (even if they consider also the draw matches).

In [12] the authors suggest that a key factor in soccer match outcome prediction lies in the successful incorporation of domain knowledge into the machine learning modeling process.

In soccer, there are other types of work concerning specific aspects of the game or player performance analysis. For example, in [34] it has been presented a model that quantifies the expected outcome of a soccer possession at any time during the possession, driven by a fine-grained evaluation of the full spatio-temporal characteristics of the 22 players and the ball. In [52] the authors try to examine who are the best players in European football, and demonstrate how the players' ratings evolve over time, using plus-minus rating. Finally, in [86] it has been proposed a weighted plus/minus metric to be used as an instrument to evaluate player performance.

3.2 Soccer Match Outcome

In this section I propose a method to analyze the game, in real-time, in order to predict match results in the context of the soccer game and it is also able to determine whether the match under analysis will be win with more or less than two goals (in order to provide a more fine-grained prediction).

In a nutshell, exploiting super-visioned machine learning algorithms, I build two models: the first one to abstract the win or loss of a match, while the second one able to model the number of goals scored by the winning team. I consider a feature set related to characteristics obtainable not only at the end of the match, but also when the match is in progress.

The study deals with two research questions:

- *RQ1*: is it possible to predict soccer match results exploiting machine learning techniques?
- *RQ2*: is it possible to predict the goal number of the winning soccer team exploiting machine learning techniques?

The chapter proceeds as follows: Subsection 3.2.1 deeply describes and motivates the proposed method; Subsection 3.2.2 contains information about the dataset used in the study; Subsection 3.2.3 illustrates the results of the experiments.

3.2.1 Method

In this section I present the proposed method depicted in Figure 3.1. It is divided in two main phases: the *Model Building* (i.e., Phase I in Figure 3.1) and the 2-step Result Prediction (i.e., Phase II in Figure 3.1).

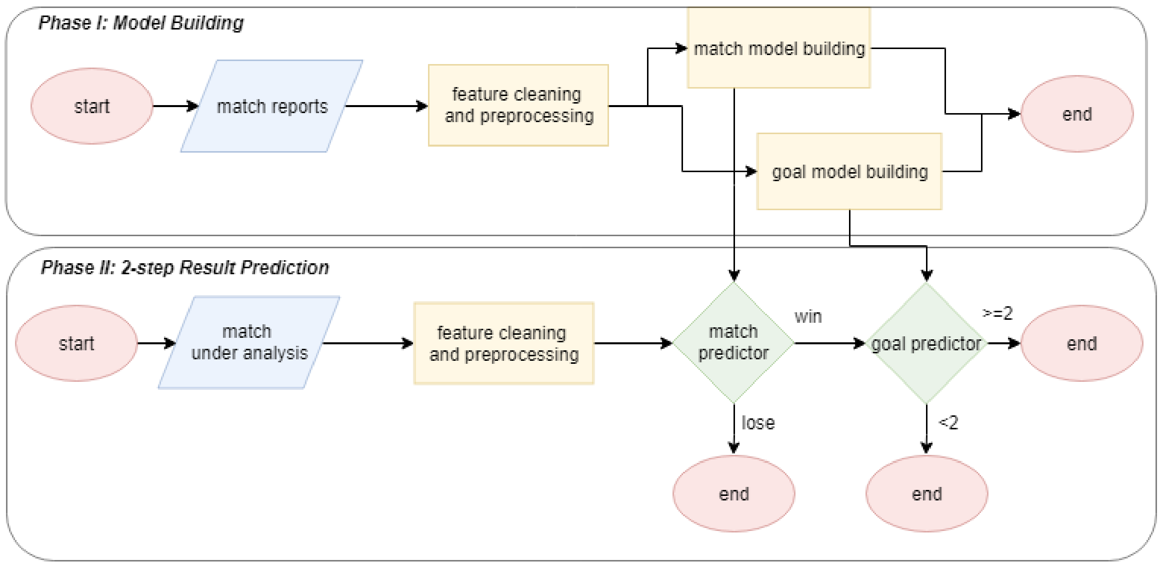


Figure 3.1: *Flowchart of the proposed method.*

The *Model Building* phase is related to the training aimed to build the two models to predict the match result and the goal number and it is composed by following modules:

- *match reports*: this module is able to data acquisition in raw format from completed matches using a plethora of information sources, for instance, digital newspapers, sport websites and RSS feed;
- *feature cleaning and preprocessing*: the aim of this module is to filter the raw data obtained in the previous step in order to extract the feature set for each match that will be considered in the two models. Basically, from the raw data for the matches, the output of this module is a Ill-formatted CVS file in which, for each examined match, the considered features appear;
- *match model building*: this module considers the feature set obtained in the previous step to build a model using the *win* or *lose* label associated to each feature vector (i.e., a match);
- *goal model building*: this module considers the feature set obtained in the previous step to build a model using the < 2 or $>= 2$ label associated to each feature vector (i.e., a match). The < 2 label is related to a match won with a number of goals

less than 2, while the $>= 2$ label is associated to a match won with a number of goals equal or greater than 2.

Once obtained the two models related to the main result of the match and to the goal number, the *2-step Result Prediction* phase has the responsibility to evaluate the results (in terms of *win/lose* and number of goals) of new matches. As a matter of fact, the features vector evaluated in the proposed method can be also used to evaluate match in progress, for instance the coach, between the first and second half, could be able to real-time predict the outcome of the match, and then think about changing the game strategy in order to win the game.

The *2-step Result Prediction* phase is composed by following modules:

- *match under analysis*: the aim of this module is the same of the first module in the *Model Building* phase: in the real-world the developed method can be also used from coaches inserting by hand the reports related to previous match or the partial result of a match;
- *feature cleaning and preprocessing*: this module is responsible to obtain the feature set from the raw information obtained in the previous step, in addition it is also able to parse the information inserted by the coach using an interface provided by the system in order to convert the information into a feature vector to input the two models built in the *Model Building* phase;
- *match predictor*: this module represents the match evaluator. It takes as input the feature vector and tests it against the model built in the *match model building* module of the *Model Building* phase (this is the reason why the inputs of the *match predictor* module are the *match model building* module and the feature vector). The output of this module is a label: *win* whether the prediction, considering the analysed feature vector, is a win of the match under analysis or *lose*, whether the prediction is a lose of the match;
- *goal predictor*: this module represents the goal evaluator. The inputs of this module are the *goal model building* module and the feature vector. Whether the *match predictor* module predicts a win of the match under analysis, the *goal predictor* module analyses the feature vector in order to predict whether the analysed match will be win with a number of goals less then two (in this case the feature vector will be marked with the < 2 label) or with a number of goals equal of greater than two (in this case the feature vector will be marked with the $>= 2$ label).

Once depicted the high-level architecture of the proposed method, I discuss in details the feature vector I considered to build the two models. Table 3.1 shows the considered features. I consider from the initial dataset only 20 features i.e., the best feature set obtained using the Best-first search principal component analysis.

I designed an experiment in order to evaluate the effectiveness of the feature vector that I propose to detect the match results and the number of goal.

The evaluation consists of classification analysis aimed at assessing whether the features are able to correctly classify between won and lose matches.

In order to perform the classification task, I selected six different classification algorithms (to improve the conclusion validity): J48, SMO, RepTree, Random Forest and MLP. For details about the algorithms the reader can refer to [98].

Table 3.1: *Features involved in the study.*

Feature	Description	Info
F1	jog_distances_km_home	path covered by the home team at low intensity exp. in Km.
F2	sprint_distance_km_away	path covered by the away team at a high intensity exp. in Km.
F3	average_speed_km_home	home team average speed exp. in Km/h.
F4	y_center_gravity_medium_1T_home	y coordinate of the gravity center of the home team, in the first half.
F5	y_center_gravity_medium_1T_away	y coordinate of the gravity center of the away team, in the first half.
F6	y_center_gravity_own_1T_away	y coordinate of the gravity center of the away team team during the attacking phase, in the first half.
F7	y_center_gravity_own_2T_away	y coordinate of the gravity center of the away team team during the defensive phase, in the second half.
F8	possession_half_away_field_percentage_home	percentage of soccer ball possession in the opposite half of the home team.
F9	possession_percentage_home	percentage of possession of the home team, during the full match.
F10	possession_half_home_field_seconds_home	time of soccer ball possession, in seconds, in their own half of the home team.
F11	possession_0_15_1T_home	time of home team possession, in seconds, from 0 minute to 15 minutes, in the first half.
F12	possession_16_30_1T_home	time of home team possession, in seconds, from 16 minutes to 30 minutes, in the first half.
F13	possession_31_45_1T_home	time of home team possession, in seconds, from 31 minutes to 45 minutes, in the first half.
F14	possession_0_15_1T_away	time of away team possession, in seconds, from 0 minute to 15 minutes, in the first half.
F15	possession_16_30_1T_away	time of away team possession, in seconds, from 16 minutes to 30 minutes, in the first half.
F16	possession_31_45_1T_away	time of away team possession, in seconds, from 31 minutes to 45 minutes, in the first half.
F17	possession_0_15_2T_away	time of away team possession, in seconds, from 0 minute to 15 minutes, in the second half.
F18	balls_recovered_midfield_right_away	number of recovered on the right of the midfield area from the away team.
F19	attacks_from_center_away	number of away team attacks from the center of the field.
F20	possession_midfield_opposing_percentage_away	midfield possession in percentage of away team

3.2.2 Dataset

The dataset has been constructed from the PDF files match report provided by the Serie A league, for each match of the season 2017-2018[1]. The dataset contains 98 attributes and 378 instances. Each instance represents a specific match in the league.

I have different types of attributes, for each team of the match, as:

- attributes that give us information about possession, in each half time;
- attributes that give us information about the spatial disposition of the player in the field - area, measured in mt^2, or the center of gravity of the team in attacking and defensive stage;
- attributes that gives us information about the goals scored and conceded;
- other types of information about the teams participating to the specific match.

The data were obtained using a Java script developed by the authors able to retrieve the required information and to create the dataset.

[1]`http://www.legaseriea.it/it`

3.2.3 Evaluation

In the follow we present the descriptive statistics in order to provide statistical evidence that the considered feature set are discriminating between lose and win matches; and the classification analysis aimed to build model able to predict real-world match results in terms of won and lose matches and number of goal of the winning team.

Figures 3.2, 3.3, 3.4 and 3.5 depict the violin plots related to a subset of the considered features (for reason of space I do not represent all the violin plots related to the feature set under analysis).

In a nutshell, the violin plots represent the distributions related to teams that have lost (represented with the lose label) and team that have won (represented with the win label): the wide horizontal part in the violin plots represent the points where there are more observations for the analysed feature.

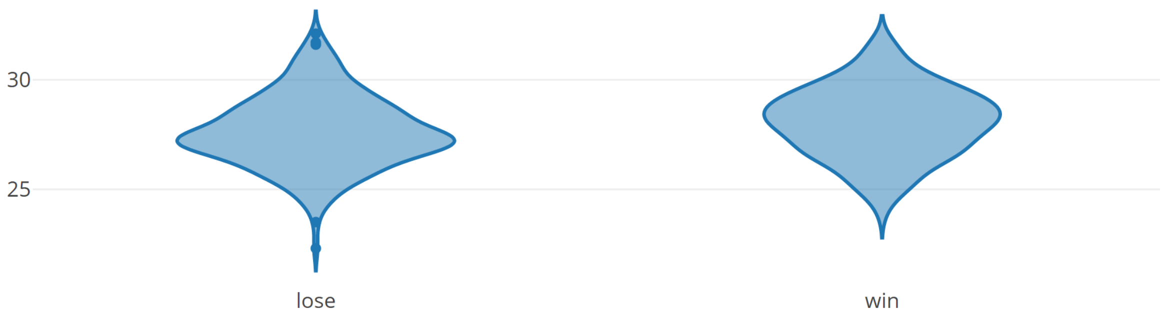


Figure 3.2: *Violin plot related to F1: jog_distances_km_home.*

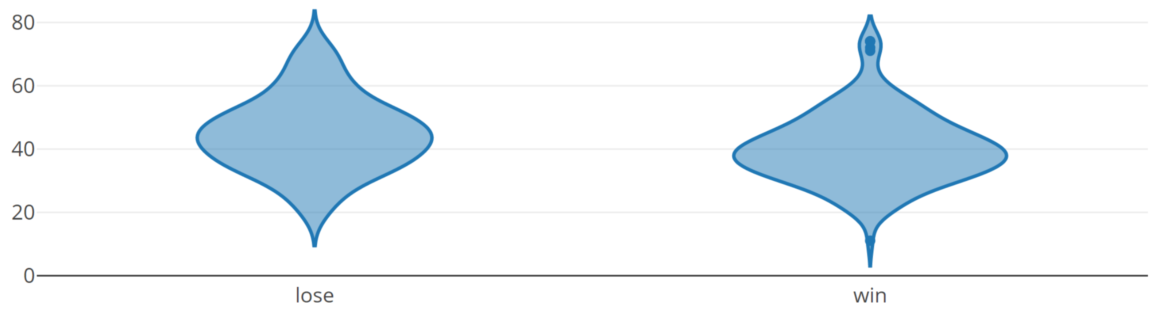


Figure 3.3: *Violin plot related to F20: possession_midfield_opposing_percentage_away.*

The violin plot in Figure 3.2 shows the distributions related to teams that have lost (represented with the *lose* label) and teams that have won (represented with the *win* label) for the F1 feature (i.e., average_speed_km_home): analysing the violin plot it seems that the teams that won the matches run through more kilometers if compared to loser teams. Anyway, the difference between the win and lose teams is not so relevant even if it is present (both of them are ranging between 25 and 30 km).

The F20 feature distributions (i.e., possession_mid-field_opposing_percentage_away) are shown in the violin plot in Figure 3.3. From distribution analysis it appears that most of the observations related to the *lose* label exhibit a slightly higher value than the ones obtained for the *win* label. This is the reason why I can state that possession during

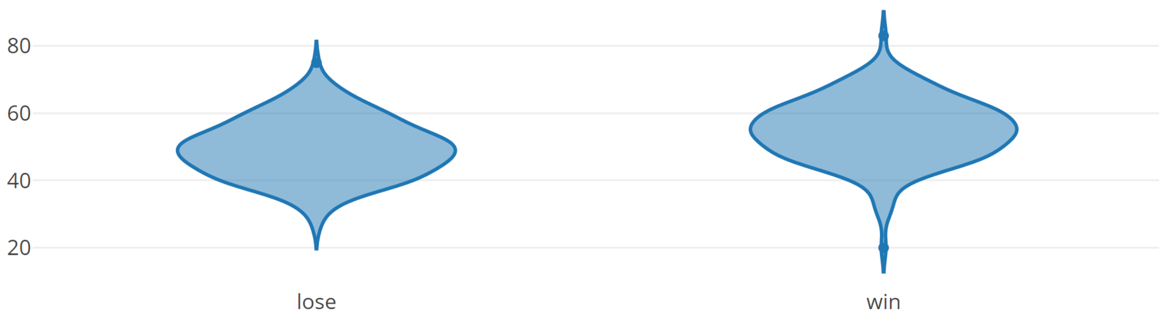


Figure 3.4: *Violin plot related to F9: possession_percentage_home.*

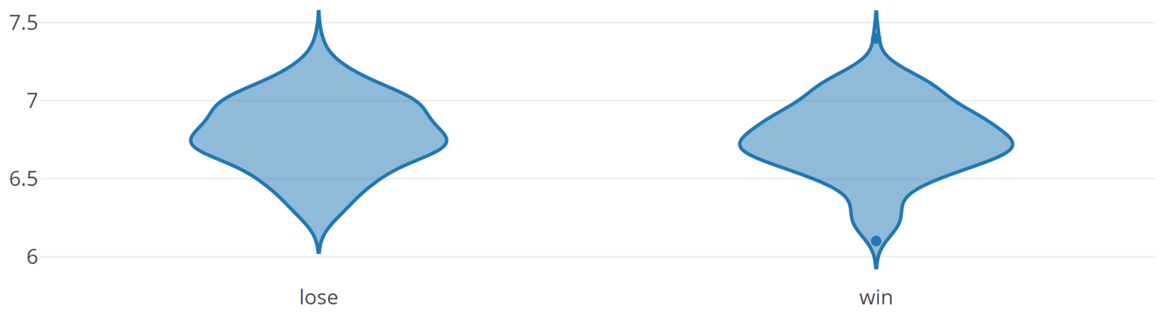


Figure 3.5: *Violin plot related to F3: average_speed_km_home.*

the match it is not directly related to the winning of a match, at least for the part of the camp of the opposing team.

The violin plot in Figure 3.4 shows the distributions for the possession_percentage_home (i.e., the F9 feature). This feature is related to possession when the team under analysis is playing at home. From the violin plot analysis it emerges that the for the F9 feature the teams that won exhibit an instance value slightly higher than the one obtains from the lose teams.

The violin plot in Figure 3.5 shows the distributions for the average_speed_km_home (i.e., the F3 feature). In this case the distributions do not exhibit particular differences between *lose* and *win* matches; as a matter of fact, the speed of players for won and lose teams is ranging between 6 and 7 km/h. The distribution related to the lose matches is slightly wider at the top compared with the *win* distribution: this is symptomatic that the *lose* teams possess more observations in the range from 7 to 7.5 km/h than the *win* one: this is symptomatic that to win a soccer match it is not relevant that the players have to run faster to the opposite team.

The classification analysis consisted of building classifiers in order to evaluate the feature vector accuracy to distinguish between *win* and *lose* matches.

For training the first classifier (the one related to the *match model building* module in Figure 3.1), I defined T as a set of labeled messages *(M, l)*, where each M is associated to a label $l \in \{win, lose\}$. For each M I built a feature vector F $\in R_y$, where y is the number of the features used in training phase ($y = 20$). The label *win* is associated to a won match, while the label *lose* is related to a lose match.

To train the second classifier i.e., the *goal model building* in Figure 3.1 I consider the a similar procedure like the one followed for the *match model building* module. In this case I defined T as a set of labeled messages *(M, l)*, where each M is associated to a label $l \in \{< 2, >= 2\}$ using the same feature vector considered in the previous model. The label < 2 is associated to a won match with a number of goal less to 2, while the label $>= 2$ is related to a match won with a number of goal equal or greater than 2.

For the learning phase, I use a k-fold cross-validation [5]: the dataset is randomly partitioned into k subsets. A single subset is retained as the validation dataset in order to evaluate the obtained model, while the remaining $k-1$ subsets of the original dataset are considered as training data. I repeated the process for $k = 20$ times; each one of the k subsets has been used once as the validation dataset [69, 87]. To obtain a single estimate, I computed the average of the k results from the folds.

The procedure is repeated two times: for the *match model building* and the *goal model building* modules.

I evaluated the effectiveness of the classification method with the following procedure:

1. build a training set $T \subset D$;
2. build a testing set $T' = D \div$T;
3. run the training phase on T;
4. apply the learned classifier to each element of T'.

In the flowchart depicted in Figure 3.1 the evaluation of the *match model building* is represented by the *match predictor* module, while the evaluation of the *goal model building* is represented by the *goal predictor* module.

Each classification was performed using 95% of the dataset as training dataset and 5% as testing dataset employing the full feature set.

As shown in Table 3.2, I obtain an average precision ranging from 0.735 (with the J48 algorithm) to 0.843 with the Random Forest algorithm. With regard to the recall, this metric in average is ranging from 0.684 (with the RepTree algorithm) to 0.842 (obtained with the SMO and the Random Forest algorithm.).

RQ1 response: the obtained results show that machine learning techniques can be able to predict soccer match results. The best performances in terms of precision and recall Ire obtained by the Random Forest algorithm, with a precision equal to 0.857 and a recall equal to 0.750 to predict a won match.

As previously discussed, for each classification I considered 95% of the dataset as training dataset and 5% as testing dataset employing the full feature set.

In order to show the performances when the training set is increasing, Figure 3.6 depicts the precision and recall trend with training set percentages ranging from 90% to 95%.

When the model is built with the 90% of the training set, both the precision and the recall are equal to 0.739. The best performances are obtained when the training set is equal to 95% (obtaining a precision equal to 0.833 and a recall equal to 0.909).

I adopted this fragmentation between training and testing dataset considering the limited number of instances in the dataset (with the aim to build a more accurate model to predict match results and the goal number) [20, 48, 64].

Table 3.3 shows the results obtained in the evaluation of the *goal predictor* module.

Table 3.2: *Classification results for match prediction: FP rate, Precision, Recall, F-Measure and Roc Area computed with J48, SMO, RepTree, Random Tree, Random Forest and Multilayer Perceptron classification algorithms.*

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
	0.375	0.750	0.818	0.783	0.841	*lose*
J48	0.182	0.714	0.625	0.667	0.841	*win*
	0.294	0.735	0.737	0.734	0.841	average
	0.250	0.833	0.909	0.870	0.830	*lose*
SMO	0.091	0.855	0.748	0.800	0.830	*win*
	0.183	0.843	0.839	0.840	0.830	average
	0.750	0.647	1.000	0.786	0.841	*lose*
Rep Tree	0.000	1.000	0.250	0.400	0.841	*win*
	0.434	0.796	0.684	0.623	0.841	average
	0.250	0.800	0.727	0.762	0.739	*lose*
Random Tree	0.273	0.667	0.750	0.706	0.739	*win*
	0.260	0.744	0.737	0.738	0.739	average
	0.250	0.833	0.909	0.870	0.955	*lose*
Random Forest	0.091	0.857	0.750	0.800	0.955	*win*
	0.183	0.843	0.842	0.840	0.955	average
	0.375	0.786	0.769	0.880	0.830	*lose*
Multilayer Perceptron	0.180	0.722	0.625	0.769	0.830	*win*
	0.292	0.754	0.697	0.833	0.830	average

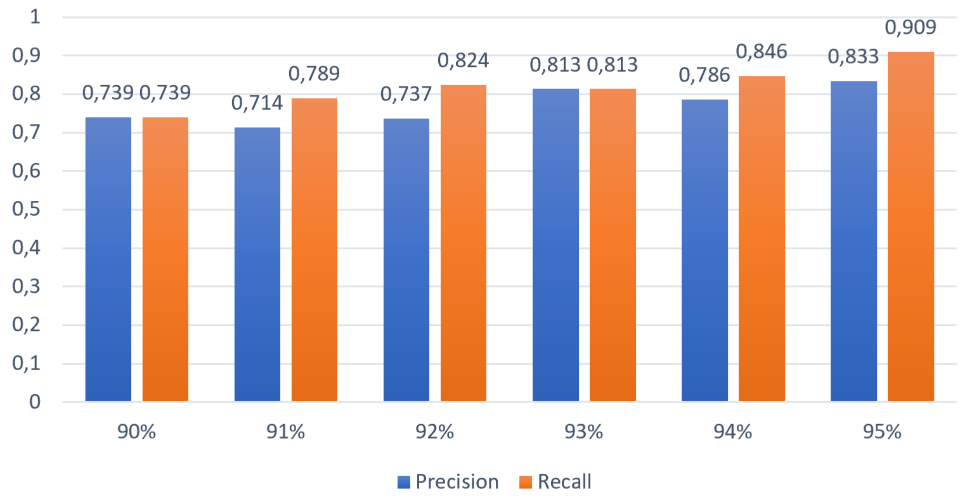


Figure 3.6: *Bar charts related to precision and recall with different training set percentages (from 90% to 95%) using the Random Forest algorithm.*

The best algorithm for goal number prediction is RandomForest, with an average precision equal to 0.862 and an average recall equal to 0.868. The less performing algorithms is J48, with an average precision equal to 0.749 and an average recall equal to 0.699. RandomForest algorithm outperforms the other algorithms since it considers a multitude of trees differently from the other considered classification approaches.

RQ2 response: the goal prediction analysis demonstrate that machine learning techniques exhibit the ability to predict the number of goals scored by the winning team. In detail, the Random Forest algorithm obtained a precision equal to 0.879 in the number

Table 3.3: *Classification results for goal prediction: FP rate, Precision, Recall, F-Measure and Roc Area computed with J48, SMO, RepTree, Random Tree, Random Forest and Multilayer Perceptron classification algorithms.*

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Goal Prediction
	0.643	0.749	0.799	0.756	0.721	< 2
J48	0.071	0.750	0.600	0.667	0.721	$>= 2$
	0.678	0.749	0.699	0.711	0.721	average
	0.500	0.873	0.967	0.921	0.733	< 2
SMO	0.033	0.800	0.500	0.615	0.733	$>= 2$
	0.402	0.860	0.861	0.856	0.733	average
	0.400	0.867	0.929	0.897	0.929	< 2
RepTree	0.071	0.750	0.600	0.667	0.929	$>= 2$
	0.836	0.842	0.842	0.836	0.929	average
	0.500	0.879	0.967	0.921	0.733	< 2
RandomTree	0.033	0.800	0.500	0.615	0.733	$>= 2$
	0.402	0.862	0.868	0.856	0.733	average
	0.500	0.879	0.967	0.921	0.733	< 2
RandomForest	0.033	0.800	0.500	0.615	0.733	$>= 2$
	0.402	0.862	0.868	0.856	0.733	average
	0.400	0.867	0.929	0.897	0.929	< 2
MultilayerPerceptron	0.071	0.750	0.600	0.667	0.929	$>= 2$
	0.836	0.842	0.842	0.836	0.929	average

of goal prediction less than two, and a precision equal to 0.8 in the number of goal prediction equal or greater to two.

3.3 Soccer Player Position

In this section, I propose an approach to predict the player positions in a soccer match that can be used to verify, also in real-time, if a specific player observes the guidelines given by the coach. Additionally, this method can be used after the match, to analyze the behaviour of the team and make considerations on several aspects to improve performances during the training; or to analyze the next opponent team in order to get some kind of information that can be used to obtain a strategical advantage before the match. Similar objectives have never been considered yet with our best knowledge. In this method, I exploit supervised machine learning techniques by considering several classification algorithms to enforce the conclusion validity. In detail, the proposed method exploits features related to the relative positions of the ball in x and y axis, other features as, for instance, the player speed.

The analysis proceeds as follows: Subsection 3.3.1 contains the proposed methodology for soccer player position detection; in Subsection 3.3.2 it has been showed the dataset used in the analysis; Subsection 3.3.3 presents the experimental results.

3.3.1 Method

I propose a methodology to identify the zone in which a specific player is located, starting from other types of information such as the position of the ball and the direction toward the player are moving and looking at. It is important to know and describe the position of a player in terms of the ball position and body orientation in order to construct a model that can be used to predict, in each specific moment and tactical situation, in which zone of the field a player should be located in. Additionally, with this approach, when there is a negative event for a specific team, such as, for example, a goal conceded, the constructed model can be used to verify if a specific player or group of players I in a wrong position, respect than the coach instructions. The important point of this methodology is that it can be applied also to real-time data, extracted from a specific time window during a game, in order to verify, for example, if a player or group of them observe the coach guidelines.

Specifically, the method that I propose is depicted in Figure 3.7 and it is characterized by 4 different steps:

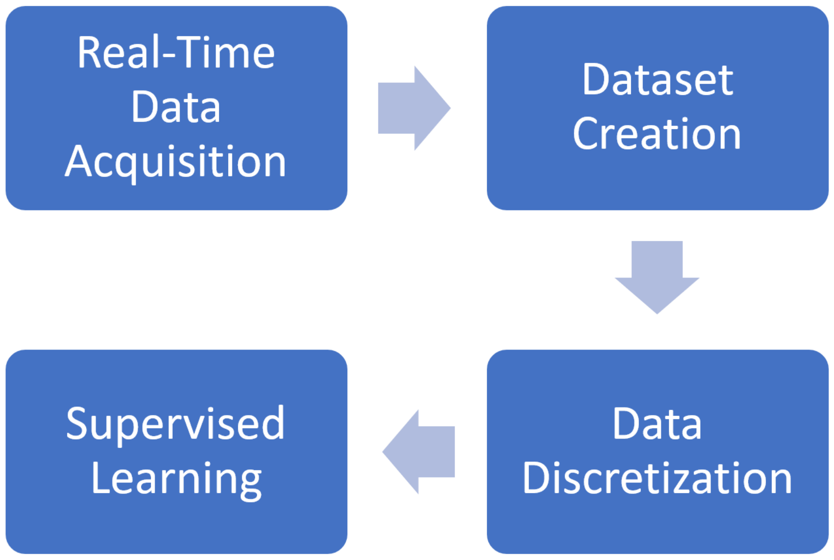


Figure 3.7: *Methodology.*

- Real-Time Data Acquisition
- Dataset Creation
- Data Discretization
- Supervised Learning

Real-Time Data Acquisition

The first step is about the data collection. The proposed methodology can consider, as source of information, video capturing method and advanced tracking system, like GPS system. These technologies allow to analyze real-time data, acquired, for example, in a specific time window - every 15 minutes.

Clearly, video capturing systems are widely used in sport environments all over the world, and a great effort has been put into building deep learning algorithms and computer systems for tracking object in videos, including sports. These types of systems allow us to collect a lot of data, but all types of algorithms and systems have their strengths and weaknesses. For example, video capturing systems are very sensitive to the lightning and environmental conditions, such as weather conditions, that are difficult to control, but, on the other hand, they provide a great amount of data to analyze.

For these reasons, it could be useful to combine different data acquisition approaches, such as video capturing method and tracking systems, in order to obtain better results.

Dataset Creation

The second step of our methodology is about the dataset construction, that contains the information that I have to use to perform our analysis. In this phase, it can be used some algorithms, like normalization techniques, that allow us to do some operations on the data to get it more understandable. At the end, the dataset has to contain information about the position of the players involved in the game, the position of the ball and information about players' body orientation and movements.

Data Discretization

After the dataset creation, a discretization operation on player position variables (x_player and y_player) has been performed. Discretization is the process that allows to transform continuous variables, models or functions into a discrete form. To do this, a set of contiguous intervals (or bins) that go across the desired range has been created. After that, each evaluation of player positions, in x-axis and y-axis, is assigned to one of these intervals.

In soccer scenario, discretizing player position variables, means that I divide the pitch into *k* equal-width zones and I assign each evaluation to one of these zones (Figures 3.8 and 3.9 represent a discretization with k=3). I have to perform the discretization operation only on the variables that I want to classify (player position variables), in order to assign, to each player, a specific label which represents the zone occupied by that player.

Supervised Learning

In the fourth step, it has been performed a classification operation. Specifically, for this purpose, it can be used different supervised machine learning techniques. The variables that I want to predict are those related to the positions of the players (x_position and y_position). In our specific case, the classification consists in the assignment of a specific zone of the pitch to each player involved in the game, based on the predictors that I have at our disposal (ball position, heading, direction and speed). In such a way, I want to understand if there exist some predictors that are very informative and discriminant in order to explain the position of a player in the field.

There are different types of classification algorithms that can be used for these purposes and, in the experimental results section, about our analysis, I have shown only those that are referred to the tree-based classification algorithms, since with these I obtain the best results.

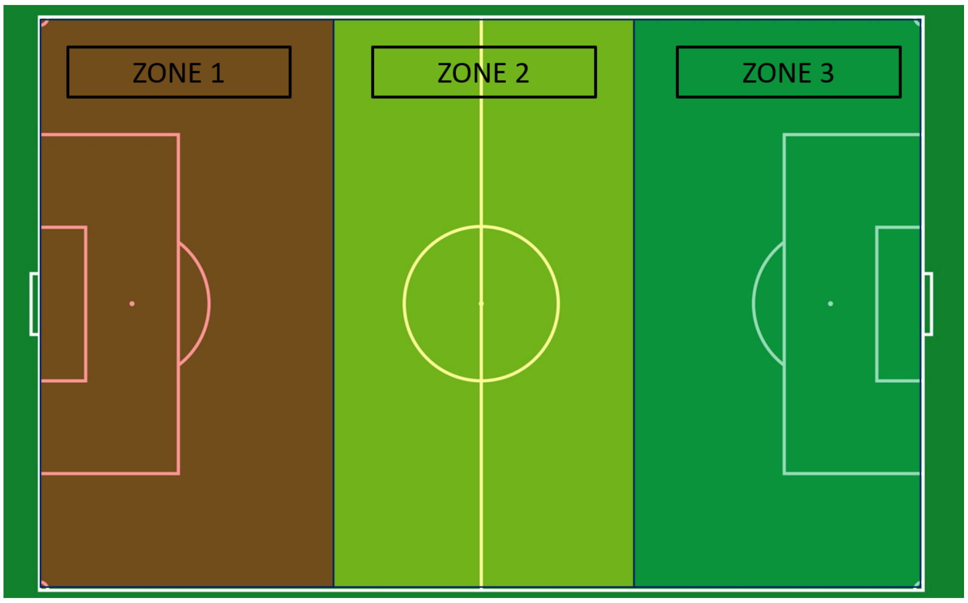


Figure 3.8: *Division of the pitch on x-axis into three zones.*

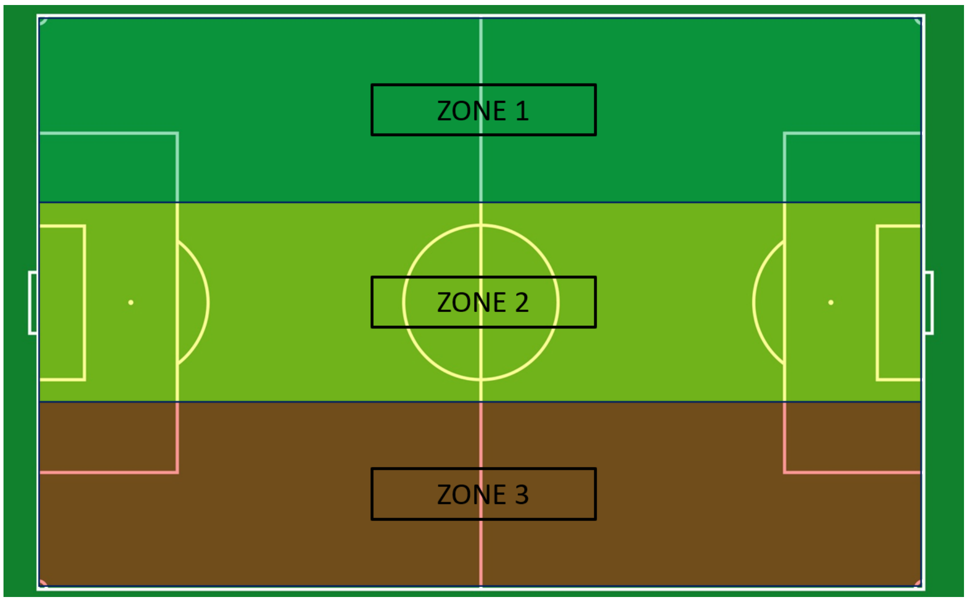


Figure 3.9: *Division of the pitch on y-axis into three zones.*

3.3.2 Dataset

For our purposes, the dataset used has been constructed in [75]. In this work, the authors have proposed a dataset of elite soccer player movements and ball position information. The dataset is captured at Alfheim Stadium, the home arena of Tromsø (Norway), during the match against Tottenham Hotspur. All the data refers to the home team. The player positions are measured at 20 Hz using the ZXY Sport Tracking system that is based on a two-dimensional positional coordinate system, inside the stadium in which the game is played. This means, that I have 20 evaluations per second for each player of the home team, for both axes of the pitch (X and Y axes).

The reference system is composed by two-dimensional positional coordinates in which the positive x-axis points to the right with respect to the camera shooting, along the side of the field; while the positive y-axis points upwards, along the short edge of the field, as shown in Figure 3.10.

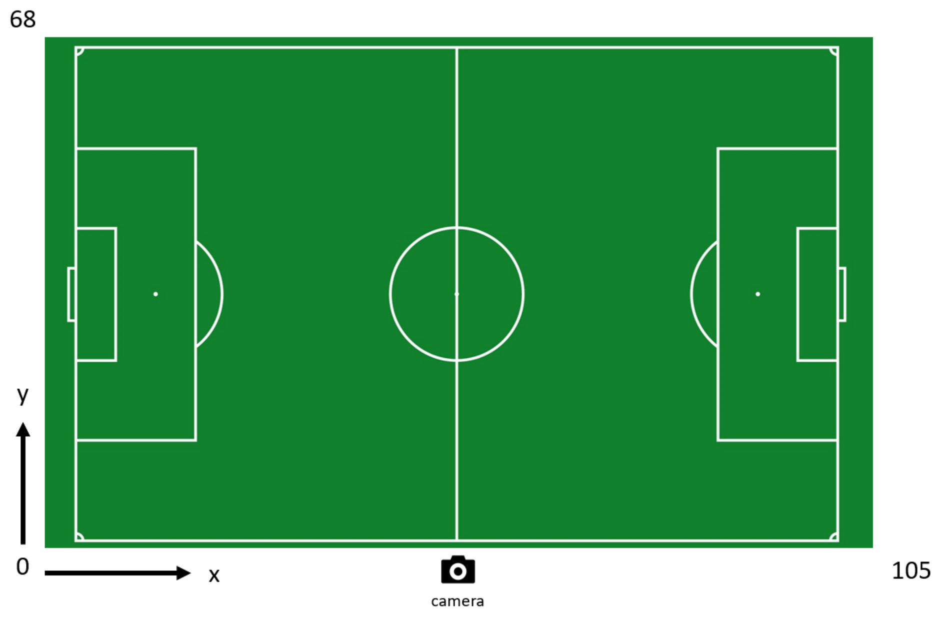


Figure 3.10: *Pitch reference system.*

The position $(0,0)$ is located in the lower-left corner of the image captured by the camera. The soccer pitch is 105 *mt* $\times$ 68 *mt* of dimension so, the values for *x_position* and *y_position* are, respectively, in the range of $0 \leqslant x \leqslant 105$ and $0 \leqslant y \leqslant 68$.

The variables referred to the players are represented in Figure 3.11.

```
'timestamp','tag_id','x_pos','y_pos','heading','direction','energy','speed','total_distance'
    ...
'2013-11-03 18:30:00.000612',31278,34.2361,49.366,2.2578,1.94857,3672.22,1.60798,3719.61
'2013-11-03 18:30:00.004524',31890,45.386,49.8209,0.980335,1.26641,5614.29,2.80983,4190.53
'2013-11-03 18:30:00.013407',0918,74.5904,71.048,-0.961152,0,2.37406,0,0.285215
'2013-11-03 18:30:00.015759',109,60.2843,57.3384,2.19912,1.22228,4584.61,8.14452,4565.93
'2013-11-03 18:30:00.023466',909,45.0113,54.7307,2.23514,2.27993,4170.35,1.76589,4070.6
    ...
```

Figure 3.11: *Samples from the 20 Hz ZXY sensor traces.*

About the ball position, instead, the information has been extracted manually by the researchers from video analysis. Merging the ball and players information, and removing some useless variables, I have obtained a dataset with approximately 495.000 observations and 7 variables, that are represented in Table 3.4.

3.3.3 Evaluation

In this subsection I present the results obtained.

Feature	Description	Info
F1	x_player	Relative position, in meters, of the player, on the x-axis.
F2	y_player	Relative position, in meters, of the player, on the y-axis.
F3	heading	Direction to the player is facing, in radians, where 0 is the direction of the y-axis.
F4	direction	Direction to the player is travelling, in radians, where 0 is the direction of the y-axis.
F5	speed	Player speed, in meters per seconds.
F6	x_ball	Relative position, in meters, of the ball, on the x-axis.
F7	y_ball	Relative position, in meters, of the ball, on the y-axis.

Table 3.4: *Predictors used in the classification analysis.*

For our analysis, I have considered three different discretizations of the variables that I want to classify, *x_position* and *y_position* of each player. Specifically, the two features have been discretized into 3, 4 and 5 labels. Each label represents a single zone with which I divide the field. So, if I consider a 3-label discretization, I are considering the pitch divided into three equal-width zones. Our goal is to assign the position of each player, in terms of x-axis and y-axis, to a specific zone.

In order to evaluate the classification that I have performed, five different metrics have been considered: false positive rate, precision, recall, F-Measure and Roc area. I have obtained, for each metrics, a value for each zone and an average value.

As previously discussed, for each classification I considered 95% of the dataset as training dataset and 5% as testing dataset employing the full feature set.

For the experimental analysis and for each type of classification, are presented the results in relation to four different tree-based algorithm: J48, Rep Tree, Random Tree and Hoeffding Tree. The analysis has been performed also using other types of algorithm, that are not illustrated since the results are really low. For each type of discretization used in the analysis, the best results are obtained in correspondence of the Random Tree algorithm; while, the worst performance, among the algorithms used, is for Hoeffding Tree.

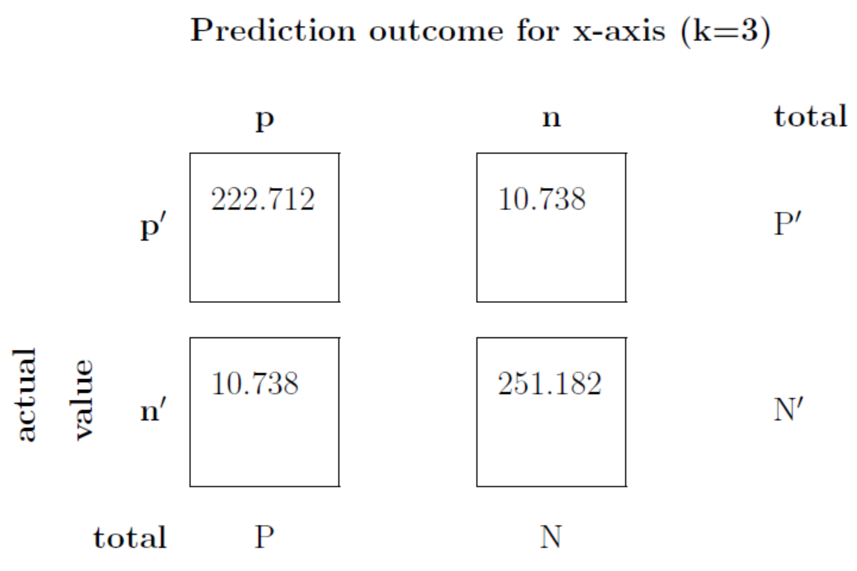
For what concern the 3-label classification, I obtain an accuracy of 0.954 for x position and 0.920 for y position. The algorithm with the worst performance is Hoeffding Tree with an accuracy of 0.781 for x position analysis and 0.668 for y position analysis.

In Figure 3.12 and Figure 3.13, it has been show the confusion matrices for Random Tree algorithm, with a 3 label discretization.

As I have said before, also for the 4-label classification, the algorithm that performs Ill is Random Tree with an accuracy of 0.936 for x position and 0.898 for y position; while, with the Hoeffding Tree, I obtain 0.678 (x-axis) and 0.583 (y-axis). The confusion matrices for Random Tree algorithm are set below (Figure 3.14 and Figure 3.15).

Finally, with the 5-label classification I have results that are very similar to the previous discretization. Specifically, I obtain 0.924 and 0.893 in relation to, respectively, x and y axis, with Random Tree. The worst performance is, even in this case, for the Hoeffding Tree algorithm with an accuracy of 0.621 (x-axis) and 0.534 (y-axis). The confusion matrices for Random Tree algorithm are set below (Figure 3.16 and Figure 3.17).

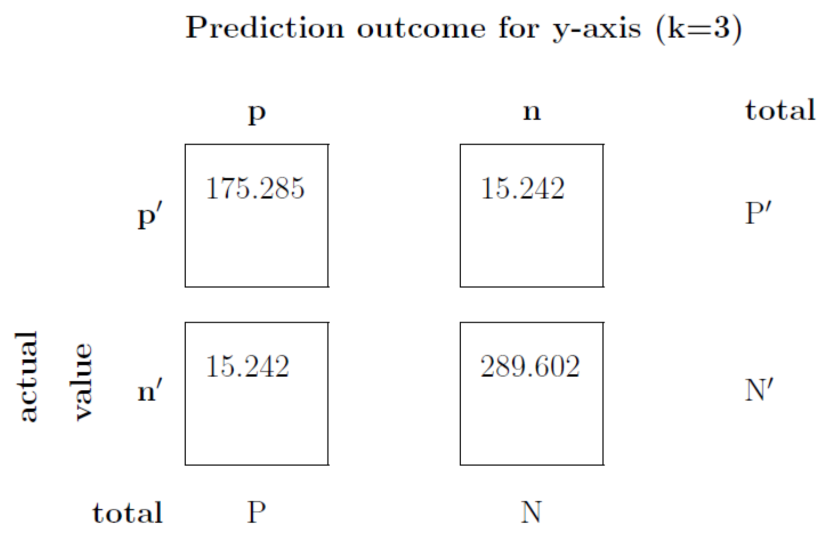
For all the analysis performed the level of accuracy of the other two algorithms, J48 and Rep Tree, is slightly lower with respect to the best algorithm, Random Tree. Another aspect of the results obtained is that, for each level of discretization, the number of FP and FN is the same, since the precision and accuracy take the same value for each



Prediction outcome for x-axis (k=3)

actual value		p	n	total
	p′	222.712	10.738	P′
	n′	10.738	251.182	N′
	total	P	N	

Figure 3.12: *Confusion matrix for Random Tree Algorithm in x-axis position prediction (k=3).*



Prediction outcome for y-axis (k=3)

actual value		p	n	total
	p′	175.285	15.242	P′
	n′	15.242	289.602	N′
	total	P	N	

Figure 3.13: *Confusion matrix for Random Tree Algorithm in y-axis position prediction (k=3).*

algorithm.

Starting from the results obtained, I can conclude that I are able, with this methodology, to identify the position of a player with a good level of accuracy. This is important because the coach can verify if a specific player, in a certain game situation, respect his guidelines in terms of position. This can be useful when the coach wants to obtain a general comprehension of the position of each player, in every game situation, based on the ball position.

Prediction outcome for x-axis (k=4)

		p	n	total
actual value	p′	205.626	14.059	P′
	n′	14.059	261.625	N′
	total	P	N	

Figure 3.14: *Confusion matrix for Random Tree Algorithm in x-axis position prediction (k=4).*

Prediction outcome for y-axis (k=4)

		p	n	total
actual value	p′	129.746	14.737	P′
	n′	14.737	336.151	N′
	total	P	N	

Figure 3.15: *Confusion matrix for Random Tree Algorithm in y-axis position prediction (k=4).*

Prediction outcome for x-axis (k=5)

		p	n	total
actual value	p′	152.574	12.549	P′
	n′	12.549	317.698	N′
	total	P	N	

Figure 3.16: *Confusion matrix for Random Tree Algorithm in x-axis position prediction (k=5).*

Prediction outcome for y-axis (k=5)

		p	n	total
actual value	p′	101.839	12.202	P′
	n′	12.202	369.126	N′
	total	P	N	

Figure 3.17: *Confusion matrix for Random Tree Algorithm in y-axis position prediction (k=5).*

Table 3.5: *Classification results for x player position (3-labeled): FP rate, Precision, Recall, F-Measure and Roc Area computed with J48, RepTree, Random Tree, Naive Bayes, Hoeffding Tree and Decision Stump classification algorithms.*

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
J48	0.007	0.901	0.885	0.892	0.966	*Zone 1*
	0.060	0.958	0.961	0.959	0.965	*Zone 2*
	0.028	0.948	0.946	0.947	0.973	*Zone 3*
	0.045	0.950	0.950	0.950	0.968	*Average*
Rep Tree	0.007	0.901	0.885	0.892	0.966	*Zone 1*
	0.060	0.958	0.961	0.959	0.965	*Zone 2*
	0.02	0.948	0.946	0.947	0.973	*Zone 3*
	0.045	0.950	0.950	0.950	0.968	*Average*
Random Tree	0.007	0.907	0.895	0.901	0.944	*Zone 1*
	0.055	0.962	0.963	0.963	0.954	*Zone 2*
	0.026	0.951	0.951	0.951	0.962	*Zone 3*
	0.041	0.954	0.954	**0.954**	0.957	*Average*
Hoeffding Tree	0.016	0.570	0.300	0.393	0.902	*Zone 1*
	0.293	0.806	0.857	0.830	0.864	*Zone 2*
	0.116	0.779	0.766	0.772	0.911	*Zone 3*
	0.213	0.781	0.788	0.781	0.883	*Average*

Table 3.6: *Classification results for y player position (3-labeled): FP rate, Precision, Recall, F-Measure and Roc Area computed with J48, RepTree, Random Tree, Naive Bayes, Hoeffding Tree and Decision Stump classification algorithms.*

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
J48	0.024	0.865	0.852	0.859	0.947	*Zone 1*
	0.078	0.908	0.919	0.914	0.943	*Zone 2*
	0.040	0.936	0.927	0.932	0.962	*Zone 3*
	0.055	0.912	0.912	0.912	0.951	*Average*
Rep Tree	0.034	0.806	0.778	0.792	0.949	*Zone 1*
	0.117	0.865	0.893	0.879	0.940	*Zone 2*
	0.052	0.916	0.893	0.904	0.965	*Zone 3*
	0.079	0.876	0.875	0.875	0.951	*Average*
Random Tree	0.023	0.876	0.872	0.874	0.925	*Zone 1*
	0.069	0.918	0.923	0.921	0.927	*Zone 2*
	0.039	0.938	0.934	0.936	0.947	*Zone 3*
	0.050	0.920	0.920	**0.920**	0.935	*Average*
Hoeffding Tree	0.022	0.589	0.174	0.268	0.806	*Zone 1*
	0.478	0.616	0.913	0.736	0.782	*Zone 2*
	0.047	0.897	0.641	0.747	0.869	*Zone 3*
	0.240	0.721	0.693	0.668	0.820	*Average*

Table 3.7: *Classification results for x player position (4-labeled): FP rate, Precision, Recall, F-Measure and Roc Area computed with J48, RepTree, Random Tree, Naive Bayes, Hoeffding Tree and Decision Stump classification algorithms.*

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
J48	0.005	0.892	0.881	0.887	0.964	*Zone 1*
	0.026	0.896	0.881	0.888	0.959	*Zone 2*
	0.082	0.945	0.953	0.949	0.957	*Zone 3*
	0.015	0.921	0.917	0.919	0.971	*Zone 4*
	0.057	0.929	0.930	0.929	0.960	*Average*
Rep Tree	0.070	0.837	0.814	0.825	0.975	*Zone 1*
	0.036	0.853	0.826	0.839	0.961	*Zone 2*
	0.121	0.920	0.935	0.927	0.957	*Zone 3*
	0.021	0.888	0.876	0.882	0.978	*Zone 4*
	0.083	0.898	0.899	0.898	0.962	*Average*
Random Tree	0.004	0.903	0.895	0.899	0.945	*Zone 1*
	0.024	0.903	0.898	0.900	0.937	*Zone 2*
	0.072	0.951	0.955	0.953	0.941	*Zone 3*
	0.014	0.925	0.922	0.924	0.954	*Zone 4*
	0.051	0.936	0.936	**0.936**	0.943	*Average*
Hoeffding Tree	0.009	0.510	0.220	0.308	0.900	*Zone 1*
	0.070	0.598	0.416	0.491	0.829	*Zone 2*
	0.455	0.736	0.855	0.791	0.806	*Zone 3*
	0.069	0.612	0.566	0.588	0.907	*Zone 4*
	0.297	0.679	0.695	0.678	0.831	*Average*

Table 3.8: *Classification results for y player position (4-labeled): FP rate, Precision, Recall, F-Measure and Roc Area computed with J48, RepTree, Random Tree, Naive Bayes, Hoeffding Tree and Decision Stump classification algorithms.*

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
J48	0.014	0.855	0.846	0.851	0.942	*Zone 1*
	0.059	0.860	0.867	0.864	0.931	*Zone 2*
	0.062	0.871	0.872	0.872	0.932	*Zone 3*
	0.018	0.955	0.949	0.952	0.977	*Zone 4*
	0.044	0.891	0.891	0.891	0.945	*Average*
Rep Tree	0.020	0.781	0.755	0.768	0.953	*Zone 1*
	0.089	0.793	0.809	0.801	0.927	*Zone 2*
	0.092	0.810	0.818	0.814	0.927	*Zone 3*
	0.024	0.940	0.920	0.930	0.982	*Zone 4*
	0.065	0.840	0.839	0.840	0.945	*Average*
Random Tree	0.013	0.867	0.861	0.864	0.924	*Zone 1*
	0.055	0.870	0.874	0.872	0.910	*Zone 2*
	0.058	0.879	0.880	0.879	0.911	*Zone 3*
	0.018	0.956	0.953	0.955	0.968	*Zone 4*
	0.042	0.898	0.898	**0.898**	0.928	*Average*
Hoeffding Tree	0.016	0.475	0.153	0.232	0.824	*Zone 1*
	0.244	0.456	0.487	0.471	0.726	*Zone 2*
	0.323	0.489	0.640	0.554	0.742	*Zone 3*
	0.017	0.947	0.747	0.835	0.927	*Zone 4*
	0.184	0.611	0.583	0.583	0.798	*Average*

Table 3.9: *Classification results for x player position (5-labeled): FP rate, Precision, Recall, F-Measure and Roc Area computed with J48, RepTree, Random Tree, Naive Bayes, Hoeffding Tree and Decision Stump classification algorithms.*

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
J48	0.003	0.889	0.879	0.884	0.962	*Zone 1*
	0.010	0.880	0.873	0.876	0.960	*Zone 2*
	0.044	0.909	0.908	0.909	0.955	*Zone 3*
	0.044	0.935	0.939	0.937	0.961	*Zone 4*
	0.011	0.913	0.906	0.910	0.966	*Zone 5*
	0.041	0.918	0.918	0.918	0.960	*Average*
Rep Tree	0.005	0.825	0.796	0.810	0.975	*Zone 1*
	0.015	0.825	0.800	0.812	0.970	*Zone 2*
	0.065	0.866	0.873	0.869	0.957	*Zone 3*
	0.077	0.908	0.913	0.910	0.965	*Zone 4*
	0.016	0.873	0.858	0.866	0.977	*Zone 5*
	0.059	0.881	0.881	0.881	0.964	*Average*
Random Tree	0.003	0.894	0.881	0.887	0.939	*Zone 1*
	0.010	0.886	0.886	0.886	0.938	*Zone 2*
	0.040	0.917	0.916	0.916	0.938	*Zone 3*
	0.050	0.940	0.944	0.942	0.947	*Zone 4*
	0.011	0.915	0.909	0.912	0.949	*Zone 5*
	0.038	0.924	0.924	**0.924**	0.943	*Average*
Hoeffding Tree	0.007	0.414	0.163	0.234	0.913	*Zone 1*
	0.026	0.491	0.290	0.365	0.872	*Zone 2*
	0.240	0.577	0.681	0.625	0.805	*Zone 3*
	0.245	0.717	0.731	0.721	-0.842	*Zone 4*
	0.048	0.540	0.441	0.485	0.909	*Zone 5*
	0.197	0.622	0.630	0.621	0.842	*Average*

Table 3.10: *Classification results for y player position (5-labeled): FP rate, Precision, Recall, F-Measure and Roc Area computed with J48, RepTree, Random Tree, Naive Bayes, Hoeffding Tree and Decision Stump classification algorithms.*

Algorithm	FP rate	Precision	Recall	F-Measure	Roc Area	Result Prediction
J48	0.010	0.845	0.840	0.843	0.941	*Zone 1*
	0.034	0.835	0.842	0.838	0.935	*Zone 2*
	0.060	0.861	0.867	0.864	0.929	*Zone 3*
	0.040	0.853	0.845	0.849	0.931	*Zone 4*
	0.010	0.972	0.968	0.970	0.986	*Zone 5*
	0.035	0.882	0.882	0.847	0.946	*Average*
Rep Tree	0.013	0.783	0.751	0.767	0.955	*Zone 1*
	0.050	0.762	0.769	0.765	0.939	*Zone 2*
	0.088	0.799	0.819	0.809	0.930	*Zone 3*
	0.057	0.789	0.775	0.782	0.936	*Zone 4*
	0.012	0.963	0.952	0.957	0.991	*Zone 5*
	0.051	0.831	0.831	0.831	0.950	*Average*
Random Tree	0.008	0.865	0.859	0.862	0.925	*Zone 1*
	0.031	0.853	0.858	0.856	0.914	*Zone 2*
	0.053	0.876	0.878	0.877	0.912	*Zone 3*
	0.038	0.862	0.859	0.860	0.910	*Zone 4*
	0.009	0.973	0.972	0.972	0.981	*Zone 5*
	0.032	0.893	0.893	**0.893**	0.930	*Average*
Hoeffding Tree	0.012	0.421	0.137	0.207	0.835	*Zone 1*
	0.064	0.374	0.186	0.248	0.737	*Zone 2*
	0.377	0.445	0.705	0.546	0.740	*Zone 3*
	0.147	0.422	0.390	0.405	0.755	*Zone 4*
	0.011	0.963	0.839	0.897	0.957	*Zone 5*
	0.159	0.558	0.549	0.534	0.803	*Average*

CHAPTER 4

Formal Methods in Soccer Analytics

4.1 State of the art

As I have said in the previous chapter, soccer represents one of the best examples of team work. Even though the success of a team is ordinarily defined in terms of its wins and losses, these wins and losses are the results of complex and intriguing interactions between the performances of individual players of the two teams and also the uncertainties pertinent to the situation. Because of this complex nature of interaction, it is difficult to decide upon the suitable set of attributes which can be used to evaluate a team behavioural style [67]. Currently, it is popular to rely on soccer experts that, typically, assign ratings to players and teams, to assess and compare their performances. There is a latent knowledge used by experts for these ratings which might be difficult to define.

Sports performance analysis enables the coach, players and managers to objectively assess and thereby improve their sporting performance. For this reason, from a strategic point of view, there is an increasing interest in this research field. In particular, team behavioural analysis, aimed to detect the team style, is attracting interest [17]. Team behaviour analysis is aimed to study and to understand the behaviour of the full team to establish a specific profile about it [28]. It has firstly been used in psychology [21] and since a few years, it has been implemented in information technology [45]. Modeling the team behaviour is not a trivial task, considering that soccer team can be considered as a complex social system [70], whose performance crucially depends on its ability to coordinate the behaviour of its members in such a way that goals are scored and the opponents' attempts to score are blocked [93]. In this context, one most interesting aspects of soccer team behaviour concerns the modelling of the ball passes between the players. While the positions and roles of players in modern football are loosely defined, the general schematic path of moving the ball is far more defined. Typically, the ball passes from the first line of defense through the midfield, where most of the passes take place, and up to the attack by players (i.e., a goal poacher) who aim to score. The fact that most ball passes take place at the midfield is almost self-evident as this is the major medium through which the ball moves from the team's first line to the forward players. This pattern can be easily observed in sites of soccer statistics[1]. Passing the ball represents a highly complex and coordinated activity that must maintain an optimal balance between players.

In the previous chapter, it has been used machine learning techniques in the sport analytics context, but with regard to these techniques, there are several well-known weaknesses. For instance, linear regression (one of the most widespread machine learning algorithms) performs poorly when there are non-linear relationships (as in the case of human behaviour modeling). They are not naturally flexible enough to capture more complex patterns. The decision tree based algorithms, another category of supervised machine learning algorithm widespread for a plethora of task, consider individual trees, prone to over-fitting because they can keep branching until they memorize the training data [54]. Moreover, machine learning based solutions lack of the so-called *explainability* [74]. The explainability is related to the important problem that complex machines and algorithms usually cannot provide insights into their behaviour and thought processes [38]. In fact, the explanability is important to ensure algorithmic fairness, identify potential bias/problems in the training data, and to ensure that the machine learning model perform as expected [32].

In order to deal with these weaknesses, in this chapter I have introduced the concept of formal methods in the context of sport analytics to verify some properties on soccer data (e.g. identify the playing style of teams).

4.2 Dunuen

In this section I propose *Dunuen*, a tool aimed to automatically generate a formal model starting from a comma separate value (CSV) file. I focus on the CSV file as source for the model building, considering that a plethora of problem are described as time-series.

Dunuen basically performs following tasks i.e., pre-processing, discretization and model building.

Furthermore, once generated the formal model of the system under analysis, *Dunuen* allows directly to dynamically invoke a formal environment verification tool (i.e., CWB-NC). A real-world case study showing the user friendly tool interfaces is presented.

The section proceeds as follows: in the next Subsection the *Dunuen* tool is presented; in Subsection 4.2.2 has been presented the architecture of Dunuen and in Subsection 4.2.3 a real-world case study is discussed.

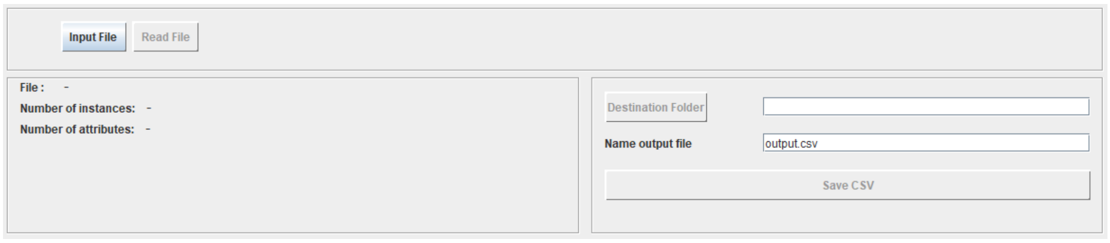
4.2.1 Technique and Implementation

This section briefly describes the *Dunuen* tool in order to explain its main functionalities available through a series of tabs i.e., *Pre-processing*, *Discretization* and *Model Building*.

[1] `whoscored.com`

Pre-processing

Dunuen provides a set of pre-processing functionalities such as loading CSV file, deleting of specific rows and saving the modified CSV file, as shown in Figure 4.1.

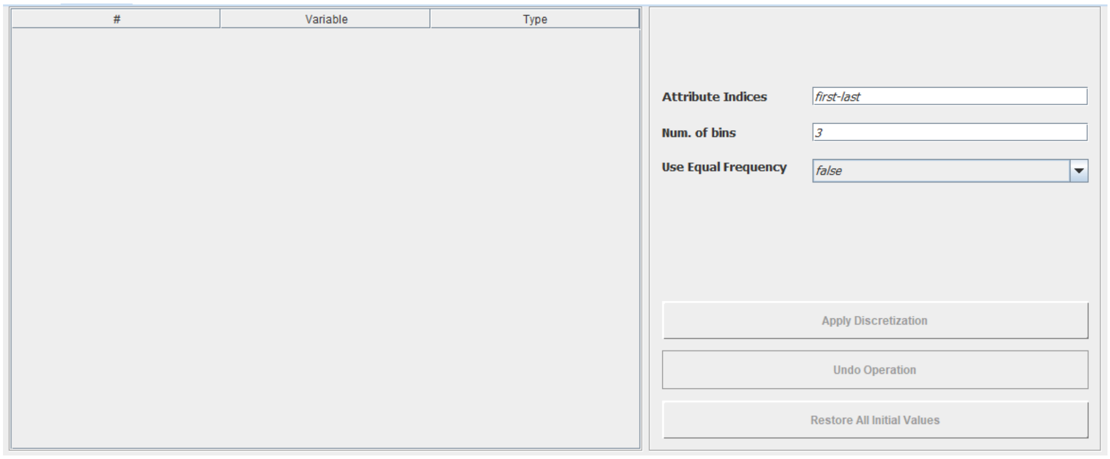


File : -
Number of instances: -
Number of attributes: -
Name output file: output.csv

Figure 4.1: *Tab of Dunuen for preprocessing operation.*

Discretization

The second tab of *Dunuen* is about the variable discretization in the loaded CSV, exploiting two different discretization methods: equal frequency or equal width method. The interface allows the user to specify the indices of the attributes to discretize, the number of the bins and the discretization method, as shown in Figure 4.2.



#	Variable	Type

Attribute Indices: first-last
Num. of bins: 3
Use Equal Frequency: false

Figure 4.2: *Tab of Dunuen for discretization operation.*

Model Building

The third tab of *Dunuen* is aimed to build the CCS model and it allows the user to:

- **Loading a .CSV file:** in this scenario the tool computes all possible permutation of the variables under analysis to build the CCS model;
- **Loading a .CCS file:** in this scenario the user can load directly a .CCS file representing the model that he/she wants to check;
- **Loading a .mu file:** through this option the user can load a .mu file containing the logical temporal properties to verify on the built CCS model.

time	$F1$	$F2$	$F3$
$t1$	$V3\_F1$	$V1\_F2$	$V2\_F3$
$t2$	$V0\_F1$	$V0\_F2$	$V3\_F3$
$t3$	$V1\_F1$	$V2\_F2$	$V3\_F3$
$t4$	...	...	...

Table 4.1: *Example of CSV file.*

P	$\overset{\text{def}}{=}$	1.	$V3\_F1.DONE \parallel V1\_F2.DONE \parallel V2\_F3.DONE;$
		2.	$V0\_F1.DONE \parallel V0\_F2.DONE \parallel V3\_F3.DONE;$
		3.	$V1\_F1.DONE \parallel V2\_F2.DONE \parallel V3\_F3.DONE;$
		4.	...;

Table 4.2: *CCS model Fragment.*

I briefly describe the model building construction: for this aim, I use Milner's Calculus of Communicating Systems (CCS) [81]. CCS is one of the most well known process algebras.

CCS contains basic operators to build finite processes, communication operators to express concurrency, and some notion of recursion to capture infinite behaviour. The semantics of a CCS process p is formally defined using the structural operational semantics. The semantic definition is given by a set of conditional rules describing the transition relation of the automaton corresponding to the behavior expression defining p. This automaton is called *standard transition system* for p. Readers unfamiliar with CCS are referred to [81] for further details. To the building model I use the following CCS operators:

- "$+$": the process $p+q$ is a process that non-deterministically behaves either as p or as q.
- ";": this operator is suitably used to express the sequentialization between two processes. The process $p;q$ means that p must terminate before the process q can start its execution.
- "$\|$": the process $p\|q$ represents the parallel execution of p and q and terminates only if both processes terminate.
- "$DONE$": is the constant that corresponds to a process whose task is to terminate without further moves.

An example of CSV containing time-series data is shown in Table 4.1, where the features are *F1*, *F2* and *F3* and I considered four discretization intervals i.e., *V0*, *V1*, *V2* and *V3*.

The CCS model describing the time series is shown in Table 4.2.

4.2.2 Overall Architecture

The typical *Dunuen* workflow comprises the following steps:

- *Model construction:* creating an abstract representation of the system;
- *Property specification:* encoding the formal specification describing the desired/-expected system behaviour;
- *Verification:* automatically verifying the correctness of the model relative to the formal specification.

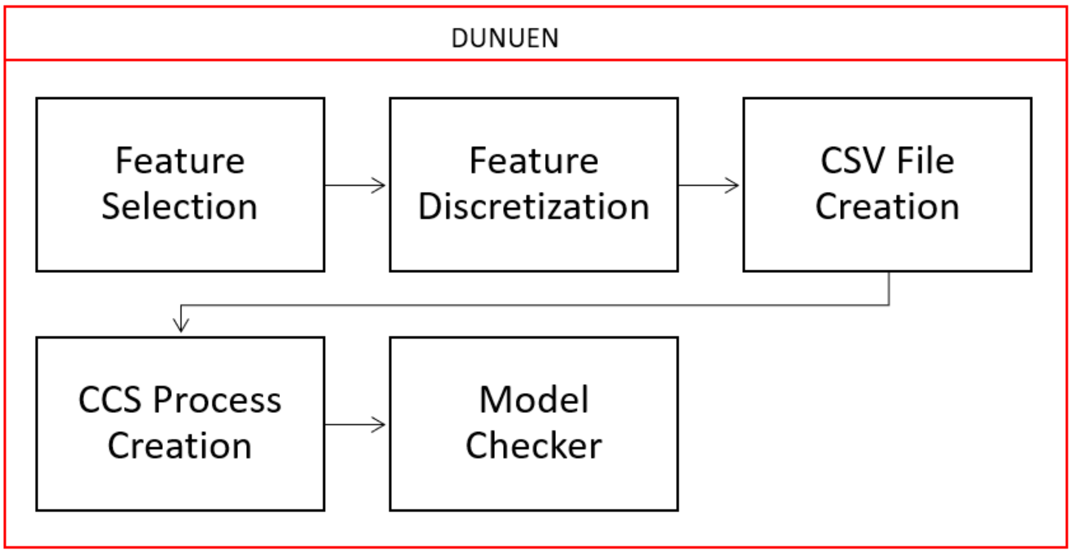


Figure 4.3: *Dunuen functionalities.*

As shown in Figure 4.3, *Dunuen* is a tool allowing the user to perform a kind of pre-processing operation, starting from a *CSV* file, as discretization or removing attributes; subsequently, it automatically creates a formal model from the pre-processed *CSV* file and, by invoking the model checker (i.e., the *CWB-NC*), and it finally verifies whether the generated model satisfies a property expressed in temporal logic through a graphic interface.

Usage

This section describes the steps that a user has to take to use *Dunuen*. I describe requirements, installation procedure and functionalities of the tool.

Requirement and Installation

Dunuen requires the following software to be installed: Java 8 or later, and the verification tool "The Concurrency Workbench of the New Century" (CWB-NC).

To install the CWB-NC on Windows platforms do the following:

1. Download cwb-nc.zip (`https://sourceforge.net/projects/cwb-nc/`);
2. Unzip the downloaded file to a temporary directory. There are many compress/uncompress windows utilities supporting the ZIP format; for example, you may check WINZIP's homepage *www.winzip.com* for more information.
3. Install the cwb-nc.
4. Download the *Dunuen* jar file.
5. Execute the command *java -jar filename.jar*.

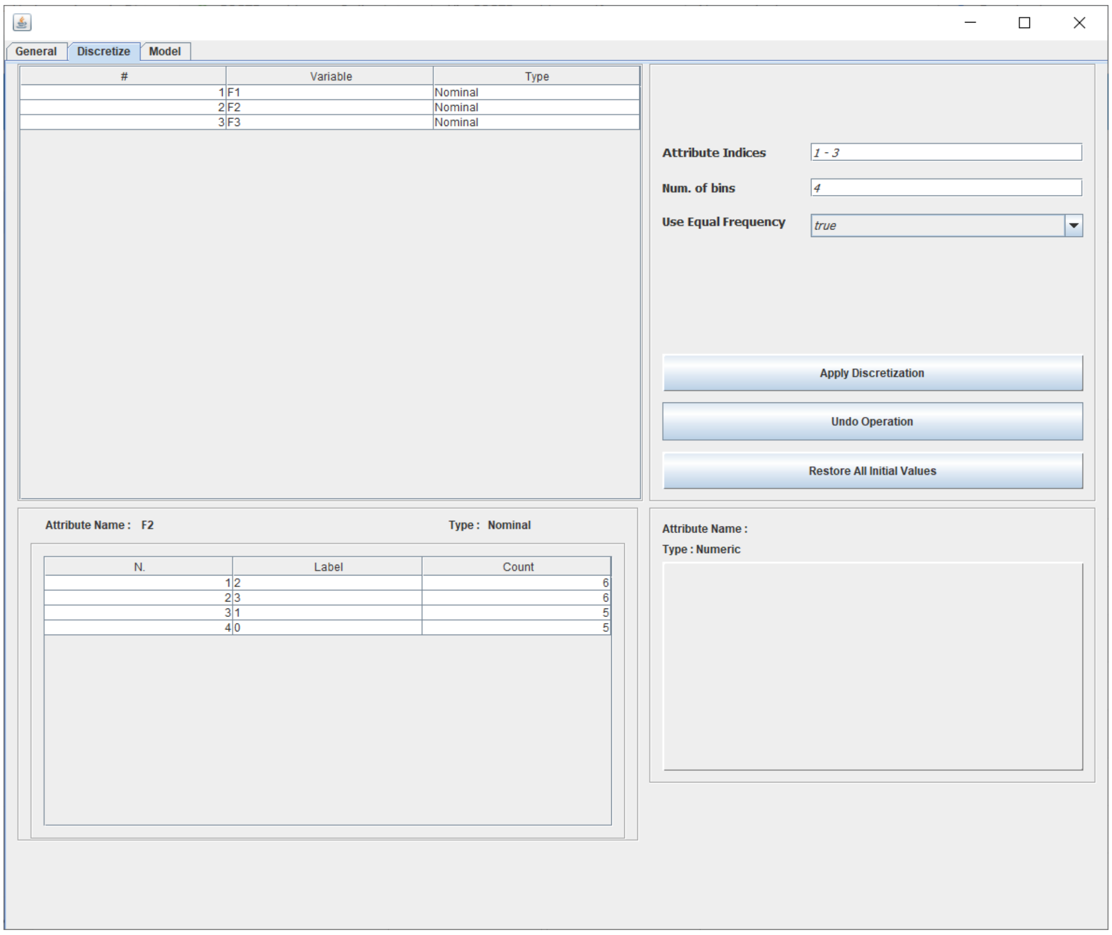
4.2.3 Case Study

In this section I present a scenario aimed to show the main functionalities provided by the *Dunuen* tool.

A real-world dataset related to a series of patients with diabetes [51, 63] whose measurement of insulin are stored pre and post breakfast, lunch, dinner and bedtime is considered. The dataset is freely available for research purpose[2]. The dataset is available in CSV file related to diabetes information characterized by 63 instances and 3 attributes. In the follow I describe the step-by-step procedure to build the CSS model with *Dunuen* and I show an example of property verification through its user interface.

Firstly, the user have to upload the CSV file and read it.

Once loaded the file, the user performs a discretization operation on the three attributes i.e., *F1*, *F2*, *F3*, using the equal frequency discretization with 4 bins, as shown in Figure 4.4. The tool allows the user to see the information related to each attribute of the file.



#	Variable	Type
1	F1	Nominal
2	F2	Nominal
3	F3	Nominal

Attribute Indices: 1 - 3
Num. of bins: 4
Use Equal Frequency: true

Attribute Name : F2 Type : Nominal

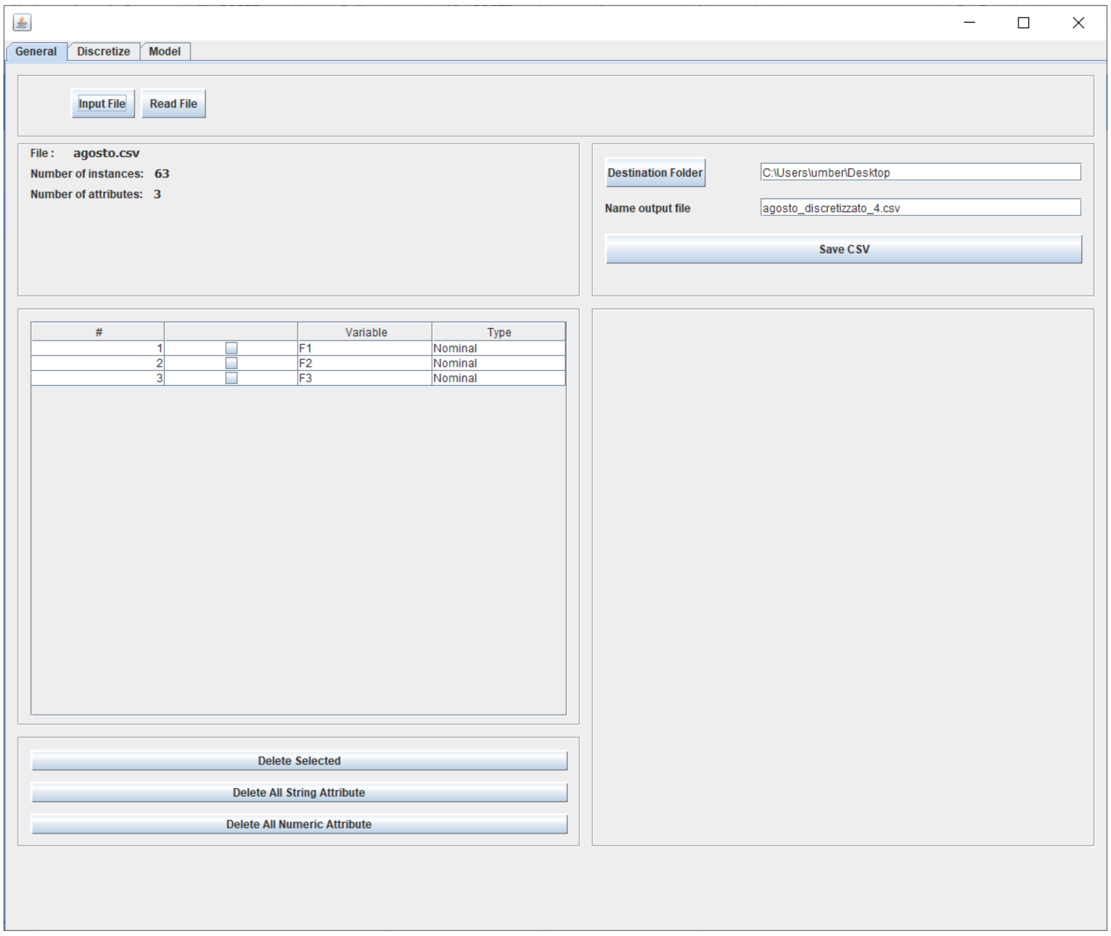
N.	Label	Count
1	2	6
2	3	6
3	1	5
4	0	5

Attribute Name :
Type : Numeric

Figure 4.4: *Example of discretization in Dunuen.*

In Figure 4.5, the user has specified the CSV file, from which the tool has constructed the CCS model, and the mu file containing the property that has to be checked on the CCS model (Figure 4.6).

[2]`https://archive.ics.uci.edu/ml/datasets/diabetes`



File : agosto.csv
Number of instances: 63
Number of attributes: 3

Destination Folder: C:\Users\umber\Desktop
Name output file: agosto_discretizzato_4.csv

#		Variable	Type
1	☐	F1	Nominal
2	☐	F2	Nominal
3	☐	F3	Nominal

Figure 4.5: *Example of loading CSV file in Dunuen.*

Once built the CCS model from the discretized CSV file, the user can evaluate a property. In this case study, I evaluate the property:

$$\varphi = \nu X.[V3\_F2]\, \texttt{ff} \wedge [-V0\_F1]\, X$$

Considering that the *F1* feature is related to *Pre-breakfast blood glucose measurement*, while *F2* is related to the *Regular insulin dose*, the property is aimed to verify that:

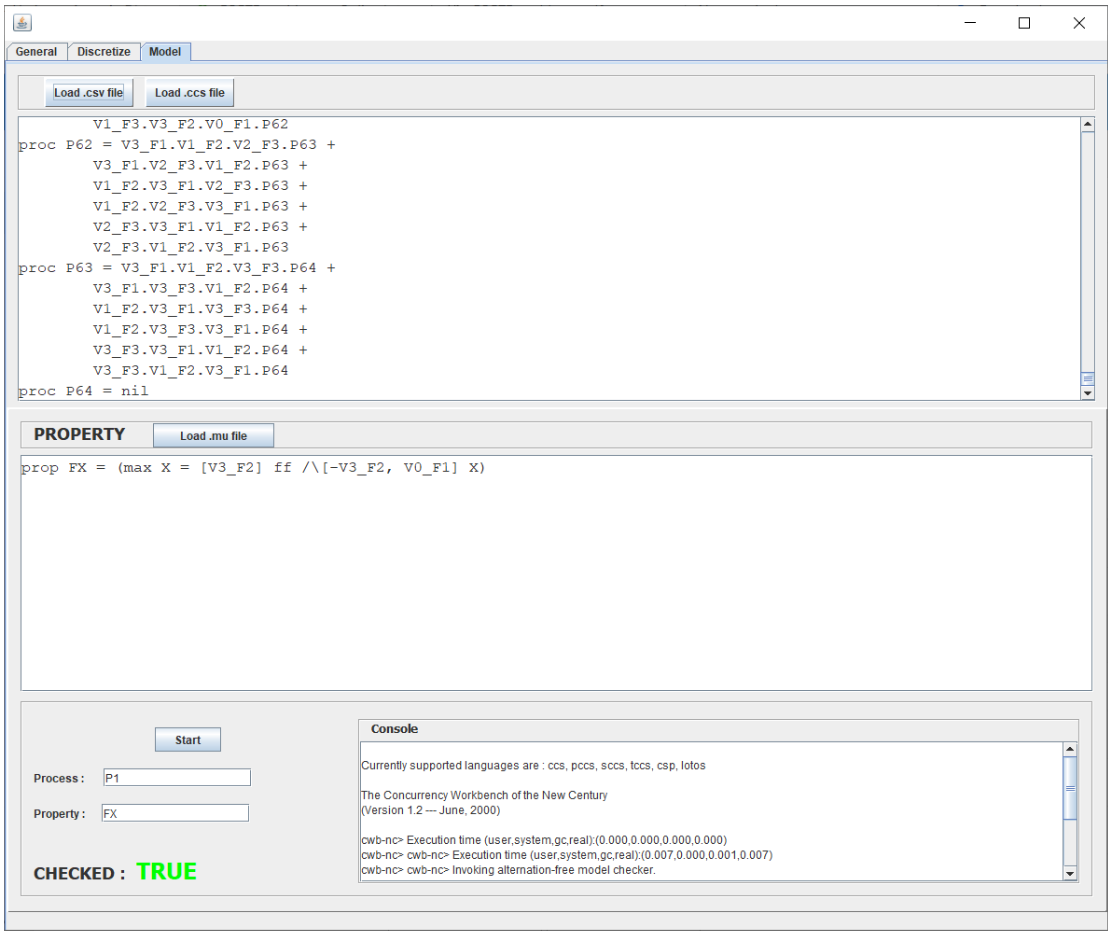
”no insulin dose has been injected (i.e., *F2* exhibits a high value) if the blood glucose pre-breakfast blood glucose measurement is low (i.e., the *F1* feature exhibits a low value)” .

In this case, the property satisfies the model, as a matter of fact the *Dunuen* tool, once verified the property by invoking the CWB-NC, outputs *TRUE*.

4.3 Real-Time Data Verification

Considering the lacking of real-time dataset in soccer environment (as I have said in the previous chapter), I propose a methodology for the creation of a dataset which can be used to verify some properties, by means of model checking, in sport scenarios.

In order to do this, object detection algorithms are very useful in several different areas. A significant effort has been put into develop algorithms and computer systems



```
        V1_F3.V3_F2.V0_F1.P62
proc P62 = V3_F1.V1_F2.V2_F3.P63 +
        V3_F1.V2_F3.V1_F2.P63 +
        V1_F2.V3_F1.V2_F3.P63 +
        V1_F2.V2_F3.V3_F1.P63 +
        V2_F3.V3_F1.V1_F2.P63 +
        V2_F3.V1_F2.V3_F1.P63
proc P63 = V3_F1.V1_F2.V3_F3.P64 +
        V3_F1.V3_F3.V1_F2.P64 +
        V1_F2.V3_F1.V3_F3.P64 +
        V1_F2.V3_F3.V3_F1.P64 +
        V3_F3.V3_F1.V1_F2.P64 +
        V3_F3.V1_F2.V3_F1.P64
proc P64 = nil
```

```
prop FX = (max X = [V3_F2] ff /\[-V3_F2, V0_F1] X)
```

Process : P1

Property : FX

CHECKED : TRUE

```
Currently supported languages are : ccs, pccs, sccs, tccs, csp, lotos

The Concurrency Workbench of the New Century
(Version 1.2 --- June, 2000)

cwb-nc> Execution time (user,system,gc,real):(0.000,0.000,0.000,0.000)
cwb-nc> cwb-nc> Execution time (user,system,gc,real):(0.007,0.000,0.001,0.007)
cwb-nc> cwb-nc> Invoking alternation-free model checker.
```

Figure 4.6: *Example of model construction and property checking in Dunuen.*

for tracking objects in videos [17], including in sports. In our case, I focus on the application of object recognition in soccer match scenes. Object detection, to be useful for this task, should be as accurate as possible and should be able to deal with a different number of objects of various sizes, partially occluded, with bad illumination and deal with cluttered scenes.

Naturally, object tracking remains a hard challenge [16]. Specifically, in outdoor situations like on a soccer field, there are different parameters that affect the robustness of the algorithms, such as the lightning, the environmental conditions, the distance of the pitch from the camera, and also the colors of the teams involved in the analysis.

4.3.1 The Method

In this section, I explain the methodology, depicted in Figure 4.7 used in this research.

Video Acquisition

The proposed method considers, as sources of information, any video acquisition from several devices (i.e., live TV, streaming, smartphone, 4K camera). Specifically, to capture the field (Figure 4.8) it has been used EVO 4K S+ camera, with a resolution of 1920x1080 pixels and a frame rate of 25fps.

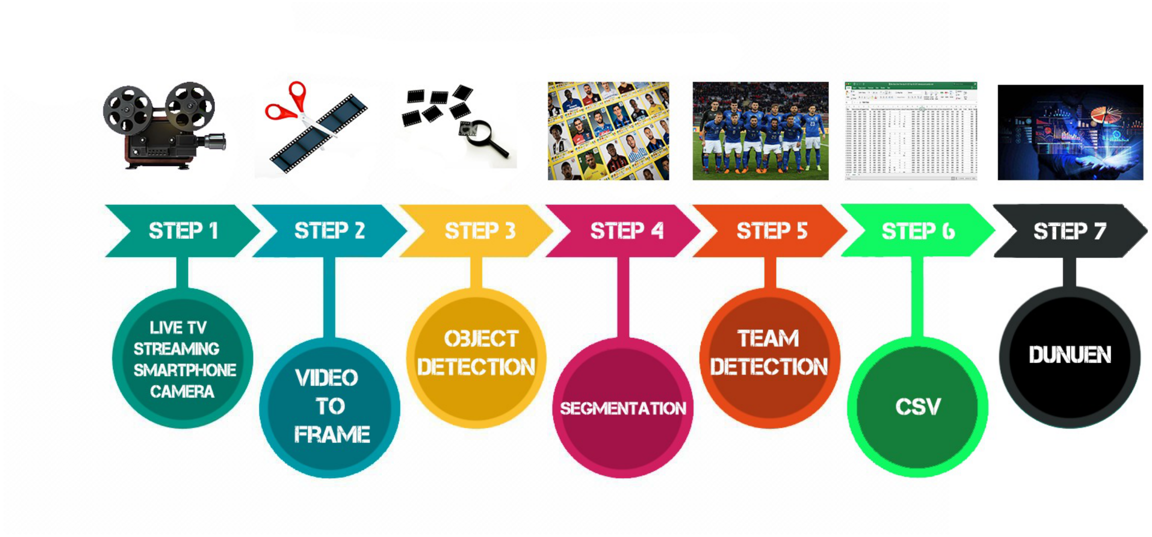


Figure 4.7: *The overall schema of the proposed methodology.*

Subsequently, before performing object detection process, I have obtained all frames from video analyzed, in order to perform the detection of player involved in the analysis, frame by frame.

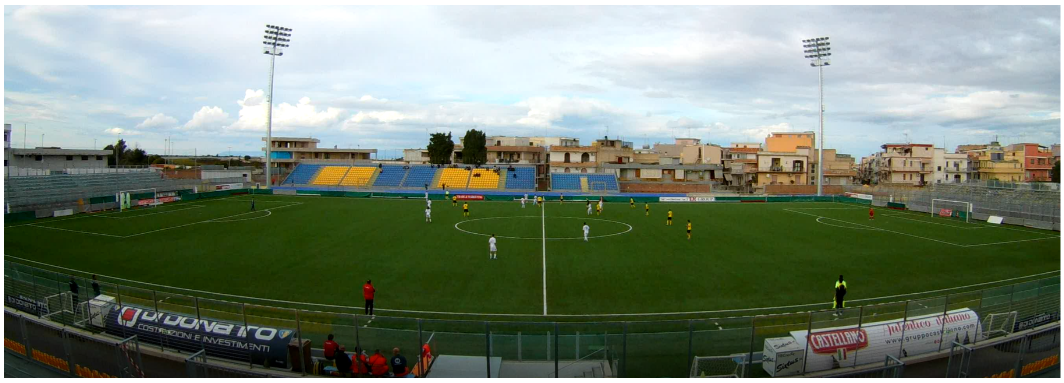


Figure 4.8: *The field view from the camera used in the study.*

Object Detection

Given an image or a video stream, an object detection model can identify which of a known set of objects might be present and provide information about their positions within the image. This step considers deep learning techniques, specifically, by invoking the Tensorflow Object Detection API, to recognize object in the frames, such as players and soccer ball. Tensorflow Object Detection API that is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. The API provides pre-trained object detection models that have been trained on the COCO data-set. COCO data-set is a set of 90 commonly found objects. In our cases I focus on persons and soccer ball, which are both part of COCO data-set.

Before team detection, I have to perform segmentation,that is the process of partitioning a digital image into multiple segments in order to simplify the representation of the frame into something that is easier to analyze. Specifically, I obtain a collection of images where each of image is an object detected in the previous step - ball or player.

Team Detection

Once I have identified the players, using the object detection API, I have to predict in which team they are. To do this, I can use OpenCV which is powerful library for image processing. OpenCV allows us to identify masks of specific colors in the object detected to know if a specific player is of the home team or the away team.

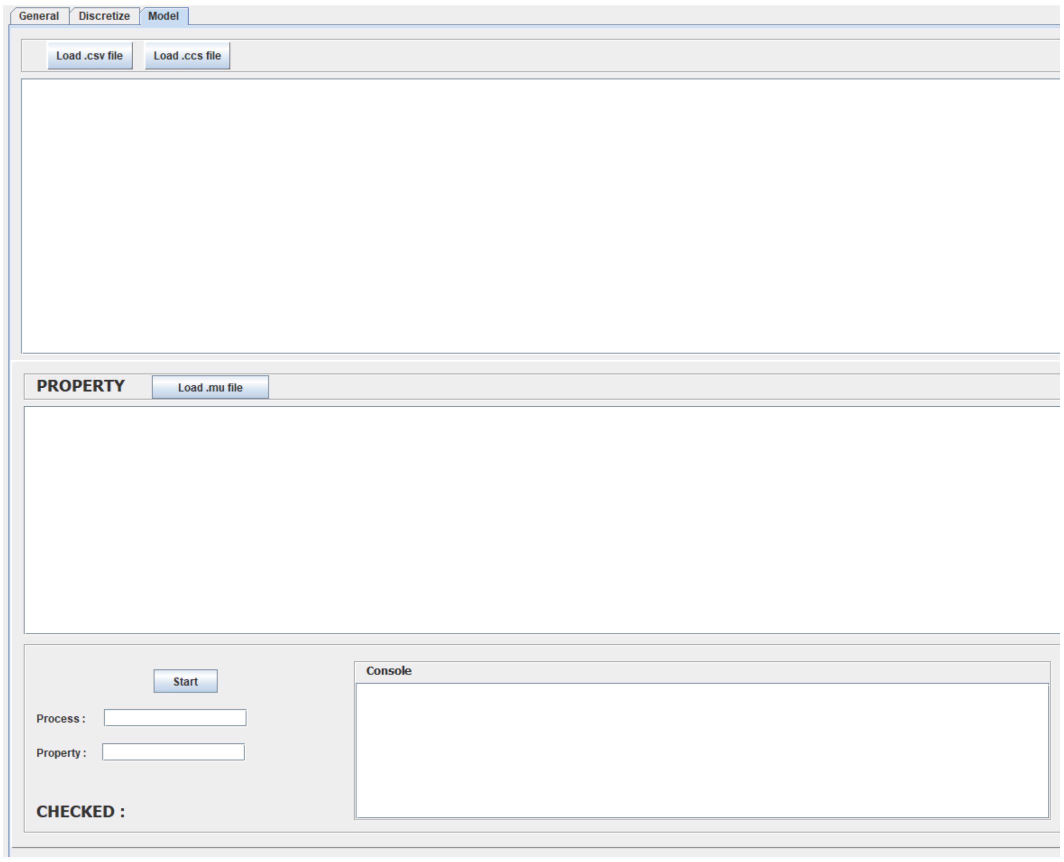


Figure 4.9: *Dunuen functionality.*

Data Format

The data obtained are store in CSV file. The file contains player position and the accuracy of the detection for each player detected in a specific frame. Each frame is represented by a row in the CSV file, as shown in Figure 4.10. For each frame, I want to identify three players per team, computing their coordinate, in terms of x-axis and y-axis in a bi-dimensional space, and the accuracy of each detection. It is important to note that the script that I have implemented in order to perform this detection, consider a specific player only if the detection accuracy is greater than 70%. If there aren't at least three player for a team in a specific frame, then the data are filled with NULL value.

```
'frame','acc_g1s1','x_g1s1','y_g1s1', ...,'acc_g1s2','x_g1s2','y_g1s2', ...
        ...
'5', '95%','1216','1096', ...,'96%','1078','874', ...
'6', '92%','411','131', ...,'97%','578','752', ...
'7', '88%','782','633', ...,'NULL','NULL','NULL', ...
'8', '98%','1113','993', ...,'86%','778','1271', ...
        ...
```

Figure 4.10: *CSV data structure.*

Model Checking

In the last step, I have used Dunuen [18] to generate a formal model [25, 83, 84] and to verify specific behavioural properties on the CSV generated in the previous step. The main functionalities of Dunuen are:

- providing a set of pre-processing functionalities such as loading CSV file, deleting of specific rows and saving the modified CSV file;
- providing variable discretization in the loaded CSV, exploiting two different discretization methods: equal frequency or equal width method;
- building of CCS model from the CSV and load the property file;
- performing model checking of the property file loaded on the constructed CCS (Figure 4.9).

4.4 Style Detection

In this section an approach based on formal methods finalised to detect the soccer team style is proposed. Additionally, our method can be designed as a decision support system used by coach and players, during a specific game, in order to understand if there exists a strategic decision that can improve the performance of a team or a player (also while the match under analysis is in progress). One of the novelty is that the designed decision support system does not hide its internal logic to the coach providing the explainability and interpretability of the results.

In this work, three different playing styles concerning spatial, tactical and bio-mechanical information have been taken into account. These information are important from the coach point of view since, knowing this aspect, it can be possible to evaluate the player performance in a specific period of time and, eventually, change the strategy of the team. Also, I itemise the distinctive points of our approach:

- I propose the adoption of formal methods in soccer analytics. This work represents the first attempt to introduce formal verification environments in this context;
- I consider a feature set obtainable while the match is in progress. This represents a novelty point with regard to the state of the art. In fact, the existing methodologies

exploit features available only at the end of matches, for instance the number of goals or the number of red cards received by the players. In details, in this work three feature categories are analysed: *tactical*, *spatial*, *bio-mechanical*;

- I model a soccer match in terms of a formal model;
- I present a set of properties aimed to detect the soccer team playing style by means of temporal logic formulae;
- I introduce the adoption of the counter-example to provide explainability and to support the coach in the tactical decision process;
- I exploit a tool, developed by authors, focused on the model generation starting from raw log in *CSV* (comma-separated values) format;
- experimental results, with a reasoned discussion, are provided with the aim to demonstrate the effectiveness of the proposed approach, compared with standard baseline approaches.

The section proceeds as follows: next subsection explains and motivates the proposed method; the *Results* subsection illustrates the results of the experiments in relation to the style considered; while, the *Discussion and Comparison* subsection provides a reasoned discussion and a case of study showing how the proposed method can be useful for supporting coach tactical decision.

4.4.1 The Method

As I have already said, sport analytics has been recently applied in sports like baseball and basketball. However, its application in soccer has been limited and only machine learning techniques have been employed, but mostly for predictions. There is a need to find out other techniques to analyze data for evaluating the playing behaviour of the team, especially in soccer which is one of the best examples of team work. There are a lot of unstructured data not fully usable by experts, although these data may contain hidden useful knowledge. Our goal is to shed light on this hidden knowledge by means of model checking which, to the authors' knowledge, has never been used before for soccer analytics. In this way, I provide a support for the coach to improve the sporting performance of the team, by analyzing tactical and spatial aspects of the team. One factor that limits the use of model checking and consequently the use of formal verification tools is mainly related to the deep knowledge of formal specification languages. For this reason I use *Dunuen* with its user-friendly interface.

The idea is to analyze behaviours performed by the players during a match, applying formal methods. The aim of this work is two-fold:

- to automate as much as possible the use of formal methods environments;
- to use model checking in soccer for analyzing data to provide a support for the coach in constructing a team, building a strategy and evaluating the playing behaviour of the team in order to improve the sport performance.

The key points are:

- to collect data concerning different features of a soccer match;

- to provide a formal data representation;
- to provide a logic representation of the behaviour of the team;
- to provide an interpretation of the collected and processed data, in order to analyze the performance of a team to improve these aspects and to carry out a coach support analysis;
- to validate our analysis with the support of an expert of the domain and of the sport analytics websites.

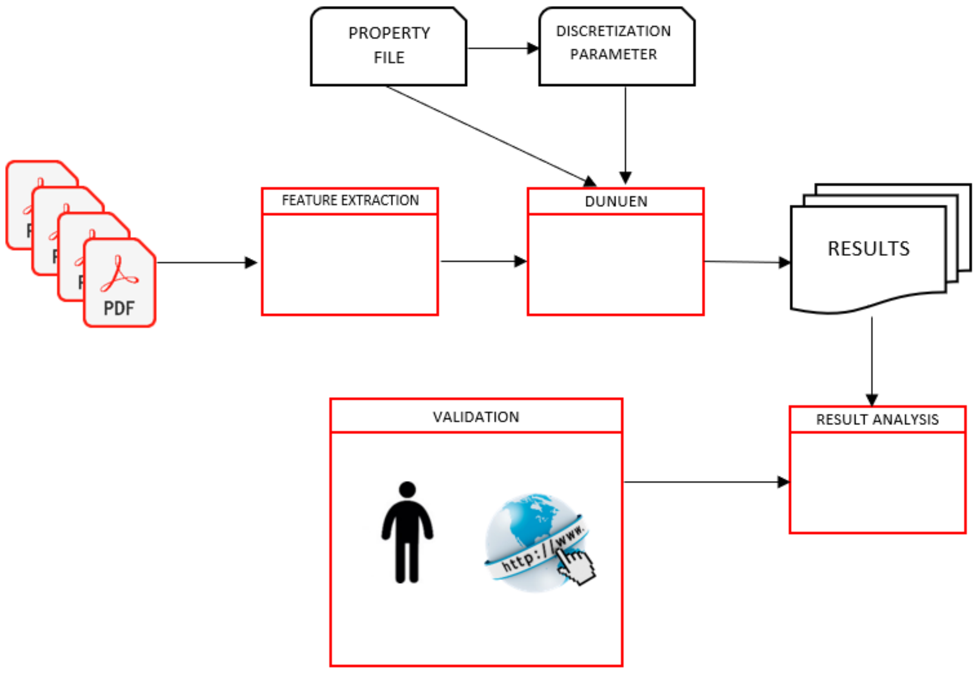


Figure 4.11: *The method.*

First I propose a method, shown in Figure 4.11, to automatically accomplish the *Model construction*, starting from data. It is worth noting that differences exist between the raw data and the insights into soccer team behaviours that an analyst would like to gain. It is important to establish relationships between the concepts characterizing behaviours and knowledge that can be extracted from data. To achieve this, data of each soccer match is considered to generate the model used for analysis, but also logic formulae are proposed to express the behavioural of a soccer team.

In details, I automatically generate a formal model starting from a *CSV* file. I focus on the *CSV* file as source for the model building, considering that soccer based repositories are usually described as time-series. I consider several team playing style. For each style, the following steps are performed:

- selection of a subset of all features of the data-set, based on the specific style;
- discretization of the selected features using a variable number of discretization classes;

- creation of a *CSV* file for each team with the discretized features;
- creation of a *CCS* process for each *CSV* file;
- definition of style properties based on the playing style that I want to verify;
- verification of the properties on the *CCS* model built in one of the previous step.

Once obtained the formal models, I will express in temporal logic the playing style team and through the model checking approach I are able to perform a deep and accurate analysis.

4.4.2 Results

In this subsection, I present our experimental evaluation. Deciding on effective team strategies and tactics is fundamental to obtain successful performance in soccer. The aim of this section is to identify and define different styles of play in Italian Soccer Championship, during the 2017-2018 season, and to classify the observed teams styles of play. For example, if soccer teams adopt an overall combination of attacking and defensive styles of play and strategy then they will increase their possibility of success. A style of play is defined as the general behaviour of the whole team to achieve the attacking and defensive objectives in the game.

Three different styles of play have been considered in order to understand if a team have a specific characterizing style. The three styles are:

- *Style 1:* this style characterizes a team on spatial information, using as feature, center of gravity (in terms of y-axis) and area information.
- *Style 2:* this style characterizes a team on tactical information, such as possession percentage.
- *Style 3:* this style characterizes a team on bio-mechanical information, using as feature the covered distance, average speed and the covered distance at different intensity level.

In our analysis, not all the styles are referred to the whole match. For example, the features considered in the Style 1 are about only the first time, in order to predict the behaviour of the team in the second time. On the other hand, Style 2 and Style 3 are considered in the entire match, since our goal is to evaluate the playing style team during the game. To express all the above styles we use the mu-calculus logic. In particular, the generic formula:

$$\varphi_1 \quad = \quad \mu X.\langle \alpha_1, \beta_1 \rangle \, \mathtt{tt} \wedge \langle \alpha_2, \beta_2 \rangle \, \mathtt{tt} \vee \langle -\alpha_1, \beta_1, \alpha_2, \beta_2 \rangle \, X$$

tells the Model Checker to find the pattern composed by either the action α_1 or the action β_1 followed by either by the action α_2 or the action β_2 This pattern could be present starting from the initial state of the model or starting from an intermediate state. In the latter case, verification is possible thanks to the recursive operator, i.e., $\vee\langle -\alpha_1, \beta_1, \alpha_2, \beta_2 \rangle X$.

Dataset

The dataset has been constructed by extracting information from a set of PDF files match report related to the Italian Soccer Championship, for each match of the 2017-2018[3] season. The data-set contains a set of 98 attributes and 378 instances. Each instance represents a specific match in the league. I have different types of attributes, for each team of the match, as:

- information about possession, in each half time;
- information about the spatial disposition of the player in the field - area, measured in mt^2, or the center of gravity of the team in attacking and defensive stage;
- information about the goals scored and conceded;
- other types of information about the teams participating to the specific match.

The data were obtained using a Java script developed by the authors able to retrieve the required information and to generated the data-set in *CSV* format.

Style 1

About the style 1, I consider features on center of gravity and area of the team, in both different phases, possession and non possession. Specifically, two property files have been constructed:

- in the first property file, the features are the longitudinal center of gravity, in possession and not;
- in the second property file, instead, in addition to the features used in the first property file, I have taken into account the area covered by the team, in both phases.

With the features used in this style, I try to understand if a specific team is an aggressive or offensive team. In order to do this, I have to highlight that the center of gravity and the area in soccer are related with the concept of pressing, which is a soccer feature that is an important indicator of the playing style of a specific team.

Pressing is a collective tactical action (i.e., carried out by more than one player), performed in situations of non-possession (also called the *defense phase*). The purpose of pressing is to close up the spaces and playing time for the team that is in possession, making it difficult for it to develop its attacking moves and easier for the other team to regain the ball. One of the most widely classifications used to define the pressing type carried out by a team refers to the part of the field where collective tactics are applied in a most systematic way. I often speak, therefore, of ultra-offensive, offensive or defensive pressing, depending on where the active attempt to disturb the opponent's play begins.

Instead, about the area, usually the more defensive team tends to have a low area, both in possession phase and in not-possession phase, compared with a more offensive team. For example, during the defensive stage, low-quality team tends to defend near to its own goal and the players are close to each other, so this means that the area covered by the team is small.

[3] `http://www.legaseriea.it/it`

Style 1 (Center of Gravity Information)

As discussed above, the variables used to characterize this style are shown in Table 4.3. After different experiments, the features have been discretized in 5 different levels: *VeryLow*, *Low*, *Medium*, *High* and *VeryHigh*.

Feature	Description	Info
F1	y_center_gravity_opponent_possession_1T	y coordinate of the gravity center when the opponent is in possession, in the first half.
F2	y_center_gravity_our_possession_1T	y coordinate of the gravity center when the team is in possession, in the first half.

Table 4.3: *Features used in the property file.*

As stated above, I want to verify if the team has a defensive playing style during a game. I have discretized the variable in 5 levels, this corresponds to divide the pitch into five equals zone. Our objective is to verify if a team have a defensive pressing, so I want check when the center of gravity variable is equals to VeryLow and Low (i.e, the actions: VeryLow_F1, Low_F1, VeryLow_F2, Low_F2 in the following formula).

This property can be expressed using the mu-calculus logic as follows:

$$\begin{aligned}\varphi_1 &= \nu X.\,[Giornata1]\,\varphi_2 \wedge [-]\,X\\ \varphi_2 &= \mu X.\,\langle VeryLow\_F1, Low\_F1\rangle\,\mathtt{tt} \wedge \langle VeryLow\_F2, Low\_F2\rangle\,\mathtt{tt} \vee\\ &\quad \langle -VeryLow\_F1, Low\_F1, VeryLow\_F2, Low\_F2, Giornata2\rangle\,X\end{aligned}$$

This property has been checked using the *Dunuen* tool, obtaining the results shown in Table 4.4. The first column of the table represents the team involved in the analysis, the second column (resp. third column) indicates the number of times in which the property is satisfied (resp. not satisfied). Clearly, if I wanted to verify if a team have an offensive attitude, I would have to construct the dual property in which the features related to the center of gravity should have a value equals to High or VeryHigh.

It is worth noting that the number of total games, for each team, is equal to 38. With the support of a domain expert and analyzing the experimental results, a threshold has been defined in order to establish the defensive behaviour of a specific team. This threshold has been set equal to 23.

Thus, I can conclude that Benevento (30/38 true), Verona (29/38 true), Udinese and Bologna (28/38 true), Cagliari, Chievo, Crotone and Genoa (25/38 true), Spal (24/38 true) and Torino (23/38 true) have the most defensive behaviour during the season. On the contrary, Napoli is the team with the most attacking style of play during the season, in terms of team pressing, since for 35 matches out of 38 do not have a *Low* center of gravity, in the two different phases. Also Roma and Milan can be considered offensive teams, while the remaining teams do not have a specific characterizing behaviour during the most of the league.

It is very interesting to note that the way in which the teams play is only a tactical and strategical choice, not directly affecting the result of a match. In fact, there is no clearly correlation between a specific tactical behaviour and the win of a championship, since, as I can note from the results obtained, Juventus, the championship winner, does not have a very offensive behaviour during the season, since the property results satisfied for 17 times out 38. So, Juventus doesn't have a unique pattern of play, in terms of

Team	True	False
Benevento	30	8
Verona	29	9
Bologna	28	10
Udinese	28	10
Cagliari	25	13
Chievo	25	13
Crotone	25	13
Genoa	25	13
Spal	24	14
Torino	23	15
Fiorentina	19	19
Juventus	17	21
Sassuolo	17	21
Sampdoria	15	23
Lazio	14	24
Inter	12	26
Atalanta	11	27
Milan	8	30
Roma	7	31
Napoli	3	35

Table 4.4: *Results obtained using center of gravity information.*

pressing. Although Juventus does not have a clearly tactical strategy during the entire of season, I can affirm that the ranking position, is correlated with the property constructed. In fact, in average, teams for which the property is less satisfied are those classified in the top positions at the end of the league.

Style 1 (Center of Gravity and Area Information)

In the second case, in addition to the center of gravity information, also the covered area in both phases, possession and not possession, in the first half has been considered. The features, shown in Table 4.5, have been discretized in 7 levels: *VeryVeryLow*, *VeryLow*, *Low*, *Medium*, *High*, *VeryHigh* and *VeryVeryHigh*.

In this analysis, I have considered more levels respect than the previous case in order to obtain a more deep knowledge of the problem.

Feature	Description	Info
F1	y_center_gravity_opponent_possession_1T	y coordinate of the gravity center when the opponent is in possession, in the first half.
F2	y_center_gravity_our_possession_1T	y coordinate of the gravity center when the team is in possession, in the first half.
F3	area_opponent_possession_1T	area covered by the shape composed by the team during the non possession phase, in the first half.
F4	area_our_possession_1T	area covered by the shape composed by the team during the possession phase, in the first half.

Table 4.5: *Features used in the property file.*

In this case, the property to be verified is related to a more defensive strategy respect than the previous property, since I also consider the area information. Usually, a well defensive-team have, not only a low center of gravity, but also a low area covered in the pitch during the two phases of play, possession and not possession.

This property is expressed in temporal logic as follows:

$$\begin{aligned}
\varphi_1 &= \nu X.\,[Giornata1]\,\varphi_2 \wedge [-]\,X \\
\varphi_2 &= \mu X.\,\langle VeryVeryLow\_F1, VeryLow\_F1, Low\_F1, Medium\_F1\rangle\, \mathtt{tt} \wedge \\
&\quad \langle VeryVeryLow\_F2, VeryLow\_F2, Low\_F2, Medium\_F2\rangle\, \mathtt{tt} \wedge \\
&\quad \langle VeryVeryLow\_F3, VeryLow\_F3, Low\_F3\rangle\, \mathtt{tt} \wedge \\
&\quad \langle VeryVeryLow\_F4, VeryLow\_F4, Low\_F4\rangle\, \mathtt{tt} \vee \\
&\quad \langle -VeryVeryLow\_F1, VeryLow\_F1, Low\_F1, Medium\_F1, VeryVeryLow\_F2, \\
&\quad VeryLow\_F2, Low\_F2, Medium\_F2, VeryVeryLow\_F3, VeryLow\_F3, \\
&\quad Low\_F3, VeryVeryLow\_F4, VeryLow\_F4, Low\_F4, Giornata2\rangle X
\end{aligned}$$

Specifically, the property results true when the following conditions are verified:

- y_center_gravity_opponent_possession_1T is equals to *VeryVeryLow*, *VeryLow*, *Low*, *Medium*;
- y_center_gravity_our_possession_1T is equals to *VeryVeryLow*, *VeryLow*, *Low*, *Medium*;
- area_opponent_possession_1T is equals to *VeryVeryLow*, *VeryLow*, *Low*;
- area_our_possession_1T is equals to *VeryVeryLow*, *VeryLow*, *Low*.

Team	True	False
Verona	24	14
Chievo	23	15
Benevento	22	16
Cagliari	22	16
Udinese	22	16
Bologna	21	17
Spal	21	17
Genoa	18	20
Crotone	17	21
Sassuolo	12	26
Torino	12	26
Fiorentina	10	28
Sampdoria	10	28
Juventus	8	30
Atalanta	7	31
Inter	5	33
Lazio	5	33
Milan	3	35
Roma	2	36
Napoli	1	37

Table 4.6: *Results obtained using center of gravity and area information.*

The result are shown in Table 4.6, in which the first column represents the team involved in the analysis, the second column (resp. third column) indicates the number of times in which the property it is satisfied (resp. not satisfied). From the results, I can provide the same consideration done for the previous analysis, even if, in this last case, there is a more clear distinction between the offensive and the defensive teams.

Style 2

In the second analysis, the property style based on tactical information has been considered. Specifically, the features taken into account are related to the overall ball possession of a team and the ball possession in the opponent half of the field, as shown in the Table 4.7. In this case, the goal is to define the play mentality of a specific team in terms of ball possession and offensive style of play.
The features are discretized in 7 levels: *VeryVeryLow*, *VeryLow*, *Low*, *Medium*, *High*, *VeryHigh* and *VeryVeryHigh*. In this case, the property results verified if the value of the two variables considered are equals to *High*, *VeryHigh* and *VeryVeryHigh*.

Feature	Description	Info
F1	y_possession_percentage	percentage of ball possession during the game.
F2	y_possession_half_opponent_field_percentage	percentage of ball possession in the opponent half of field during the game.

Table 4.7: *Features used in the Style 2 property file.*

The property considered in this section is expressed in temporal logic as follows:

$$\begin{aligned}
\varphi_1 &= \nu X.\,[Giornata1]\,\varphi_2 \wedge [-]\,X \\
\varphi_2 &= \mu X.\,\langle High\_F1, VeryHigh\_F1, VeryVeryHigh\_F1\rangle\,\texttt{tt}\wedge \\
&\quad \langle High\_F2, VeryHigh\_F2, VeryVeryHigh\_F2\rangle_{\texttt{tt}} \vee \\
&\quad \langle -High\_F1, VeryHigh\_F1, VeryVeryHigh\_F1, High\_F2, \\
&\quad VeryHigh\_F2, VeryVeryHigh\_F2, Giornata2\rangle X
\end{aligned}$$

The results are shown in Table 4.8, in which the first column represents the team involved in the analysis, the second column (resp. third column) indicates the number of times in which the property it is satisfied (resp. not satisfied).
From the results, I can conclude that, according to this style of play, Napoli is the most ball-possession team, since for 35 matches out 38, the property results satisfied. Also Roma is a team that have an offensive mentality of play.

Style 3

In the third analysis, the style based on bio-mechanical information has been considered, as represented in Table 4.9. Specifically, the features taken into account give us information about the covered distance (measured in km), covered distance at different intensity levels and the average speed.

The features considered are discretized in 7 level: *VeryVeryLow*, *VeryLow*, *Low*, *Medium*, *High*, *VeryHigh* and *VeryVeryHigh*.
The property considered is expressed in temporal logic as follows:

Team	True	False
Napoli	24	14
Roma	18	20
Inter	15	23
Juventus	14	24
Milan	13	25
Atalanta	11	27
Lazio	10	28
Sampdoria	10	28
Bologna	5	33
Chievo	4	34
Fiorentina	4	34
Genoa	4	34
Torino	4	34
Udinese	4	34
Cagliari	3	35
Verona	3	35
Spal	2	36
Benevento	1	37
Crotone	1	37
Sassuolo	1	37

Table 4.8: *Results obtained using possession information.*

Feature	Description	Info
F1	y_covered_distance_km	total distance covered by the team measured in Km.
F2	y_covered_distance_run_km	distance covered by the team at medium intensity measured in Km.
F3	y_covered_distance_sprint_km	distance covered by the team at high intensity measured in Km.
F4	y_average_speed_kmh	team average speed measured in Km/h.

Table 4.9: *Features used in the Style 3 property file.*

$$
\begin{aligned}
\varphi_1 &= \nu X.\,[Giornata1]\,\varphi_2 \wedge [-]\,X \\
\varphi_2 &= \mu X.\,\langle Medium\_F1, High\_F1, VeryHigh\_F1, VeryVeryHigh\_F1\rangle\,\mathtt{tt}\wedge \\
&\quad \langle Medium\_F2, High\_F2, VeryHigh\_F2, VeryVeryHigh\_F2\rangle\,\mathtt{tt}\wedge \\
&\quad \langle Medium\_F3, High\_F3, VeryHigh\_F3, VeryVeryHigh\_F3\rangle\,\mathtt{tt}\wedge \\
&\quad \langle Medium\_F4, High\_F4, VeryHigh\_F4, VeryVeryHigh\_F4\rangle\,\mathtt{tt}\vee \\
&\quad \langle -Medium\_F1, High\_F1, VeryHigh\_F1, VeryVeryHigh\_F1, \\
&\quad Medium\_F2, High\_F2, VeryHigh\_F2, VeryVeryHigh\_F2, Medium\_F3, \\
&\quad High\_F3, VeryHigh\_F3, VeryVeryHigh\_F3, Medium\_F4, \\
&\quad High\_F4, VeryHigh\_F4, VeryVeryHigh\_F4, Giornata2\rangle X
\end{aligned}
$$

The property is verified if the fours considered variables are equals to *Medium*, *High*, *VeryHigh* and *VeryVeryHigh*. The results are shown in Table 4.10.

Team	True	False
Napoli	28	10
Inter	26	12
Udinese	18	20
Bologna	16	22
Fiorentina	16	22
Verona	16	22
Atalanta	15	23
Chievo	12	26
Crotone	10	28
Lazio	10	28
Cagliari	8	30
Benevento	7	31
Sassuolo	7	31
Torino	7	31
Milan	6	32
Sampdoria	6	32
Spal	6	32
Juventus	4	34
Roma	4	34
Genoa	2	36

Table 4.10: *Results obtained using bio-mechanical information.*

A very interesting aspect of this section is the difference, from a bio-mechanics point of view, of the first and second ranking classified at the end of the season, respectively Juventus and Napoli. Specifically, Juventus has had a great physical effort only in 4 games out 38 during the whole league, instead for Napoli 28 games out 38. Then Napoli is a particular case in Italy, about its playing style, because, as I can see from the first and the second analysed styles, it has an offensive mentality and, as I can see from the third style, it has a great physical effort during most of the league (28 games out 38).

4.4.3 Discussion and Comparison

In this subsection, a reasoned discussion about the obtained results is provided. Also a case study to demonstrate the explanibility and the interpretability of our method is presented. In particular, in order to asses our methodology, our results are compared using two different ways.

First a comparison with standard baseline approaches have been made, then also other sources of information have been used. In the following I describe both theses comparisons.

To compare our results with a baseline approach, I have applied, for each style, the cluster analysis, a widely used technique in machine learning. Cluster analysis is aimed to group a set of objects in such a way that objects in the same group (called a *cluster*) are more similar (with some notion of similarity) to each other than to those in other groups (clusters). The machine learning experiments are accomplished with the Weka tool, a very popular machine learning suite. The *Simple K Means* cluster analysis algorithm is considered, with a number of cluster equal to 2.

I briefly reconsider the three style properties previously discussed. In the Style 1, spatial information are taken into account in order to construct the style property. Specifically, I are focused to characterize a defensive playing style for a specific team. Our results, as described in the previous section, show, for example, that Napoli is the most offensive team during the season, in terms of center of gravity.

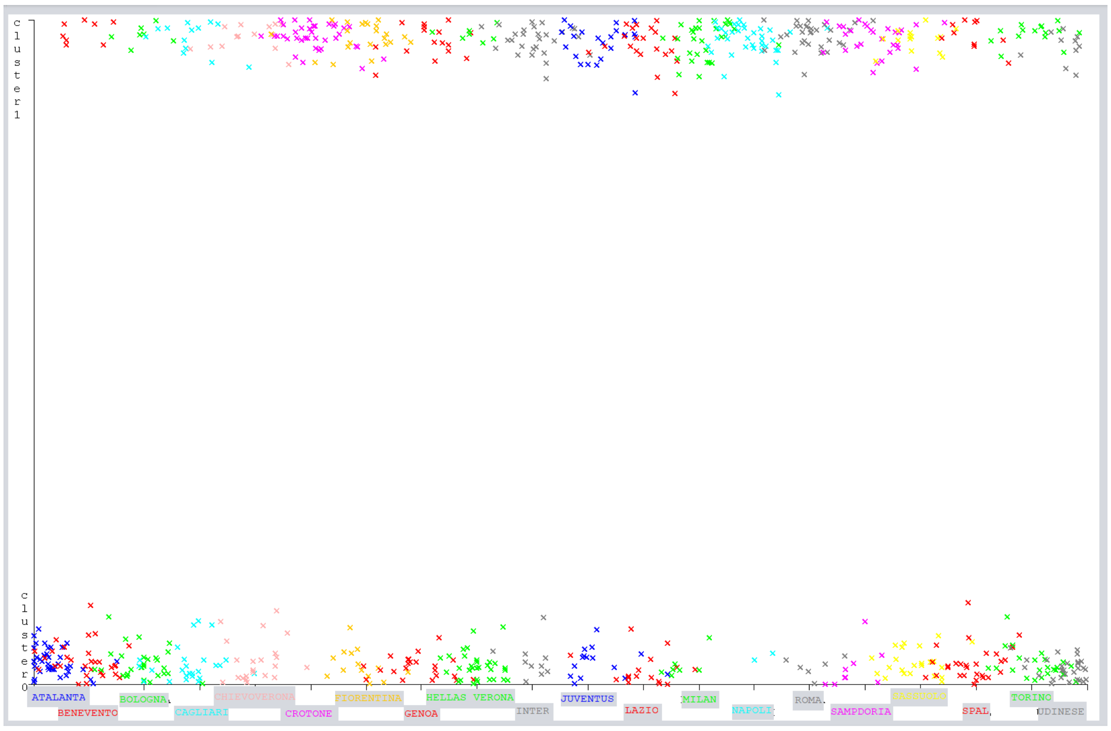


Figure 4.12: *Results obtained with cluster analysis for the Style 1.*

The comparison with a cluster analysis approach (Figure 4.12), with 2 clusters, confirms that our results are in according to the machine learning technique, since I obtain a cluster in which I have the offensive team, such as Napoli, Roma, Sampdoria, Lazio and Milan; a second cluster, in which there are Benevento, Bologna, Cagliari, Chievo, Verona and Udinese. In the second analysis of the Style 1, the cluster analysis has confirmed the same considerations done for the previous property. This is shown by the similarity of the two figures. Differently, in our formal method approach there is a more clear distinction between the offensive and the defensive teams, and this is one of the strengths of our method compared to the cluster analysis (Figure 4.13).

In the Style 2, instead, it has been used tactical information and the results show that, also in this case, Napoli is the most offensive team, followed by Roma, Inter and Juventus, in terms of ball possession. The results are in agreement with the cluster analysis performed on the same variable as shown in Figure 4.14.

In the Style 3, the features considered concern bio-mechanical information in order to understand the physical effort of a specific team during the whole league. In this case, from the results obtained by formal methods and cluster analysis (Figure 4.15), it is very interesting to note the behaviour of the first three ranked teams at the end of the season, Juventus, Roma and Napoli. Specifically, Napoli, for which the property is satisfied in 28 games out 38, is the team that has the most physical effort during the entire league. Instead, Juventus and Roma (4 games out 38), except Genoa (2 games out 38) have the least effort during the league. From a general point of view, it seems that there is no clearly relationship between this type of information and the ranking position at the end of the league. All these results are confirmed by the cluster analysis,

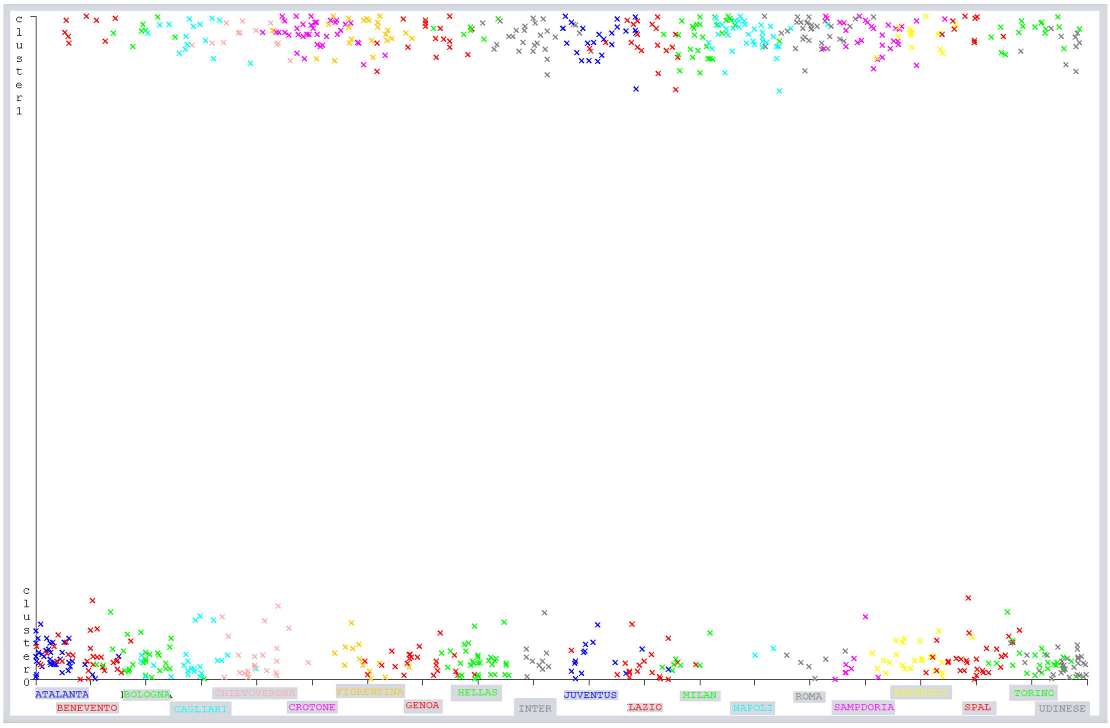


Figure 4.13: *Results obtained with cluster analysis for the Style 1 with area information.*

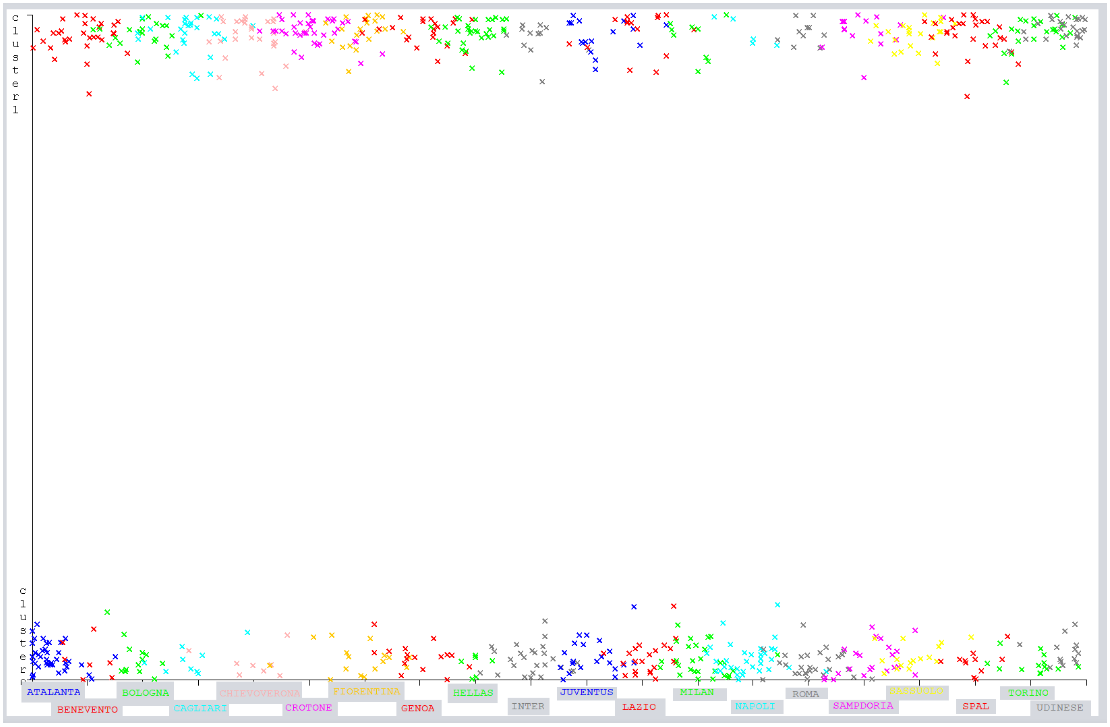


Figure 4.14: *Results obtained with cluster analysis for the Style 2.*

since I can see that Napoli is mostly present in the first cluster, while Juventus, Napoli and Genoa are mostly in cluster 2.

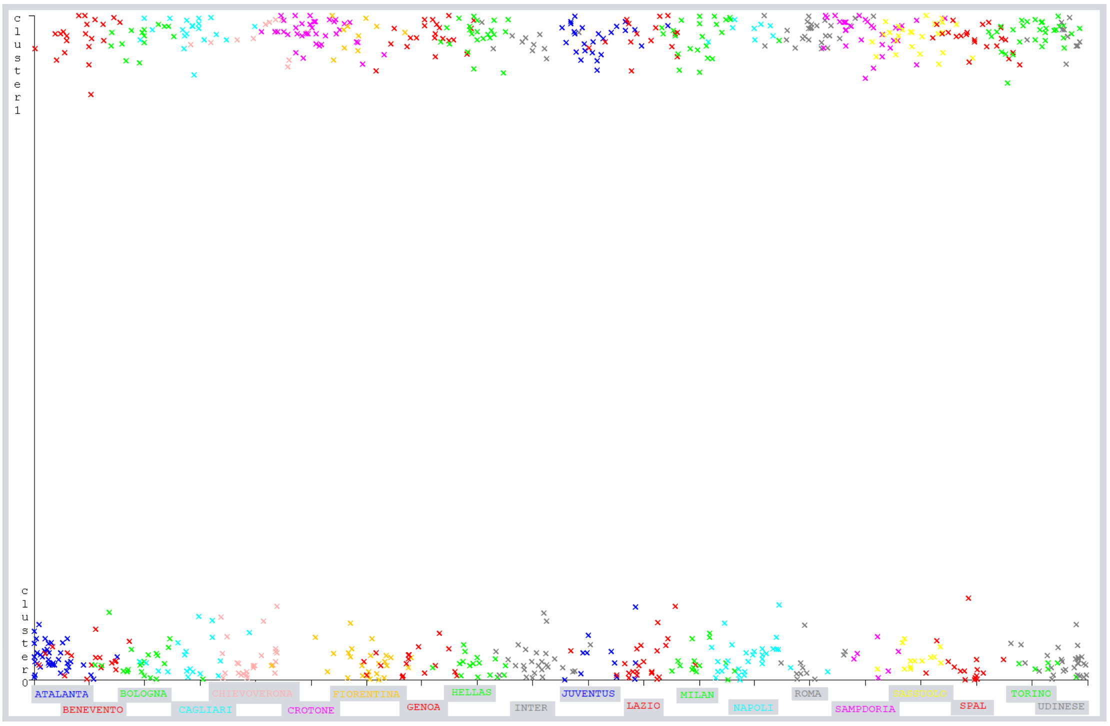


Figure 4.15: *Results obtained with cluster analysis for the Style 3.*

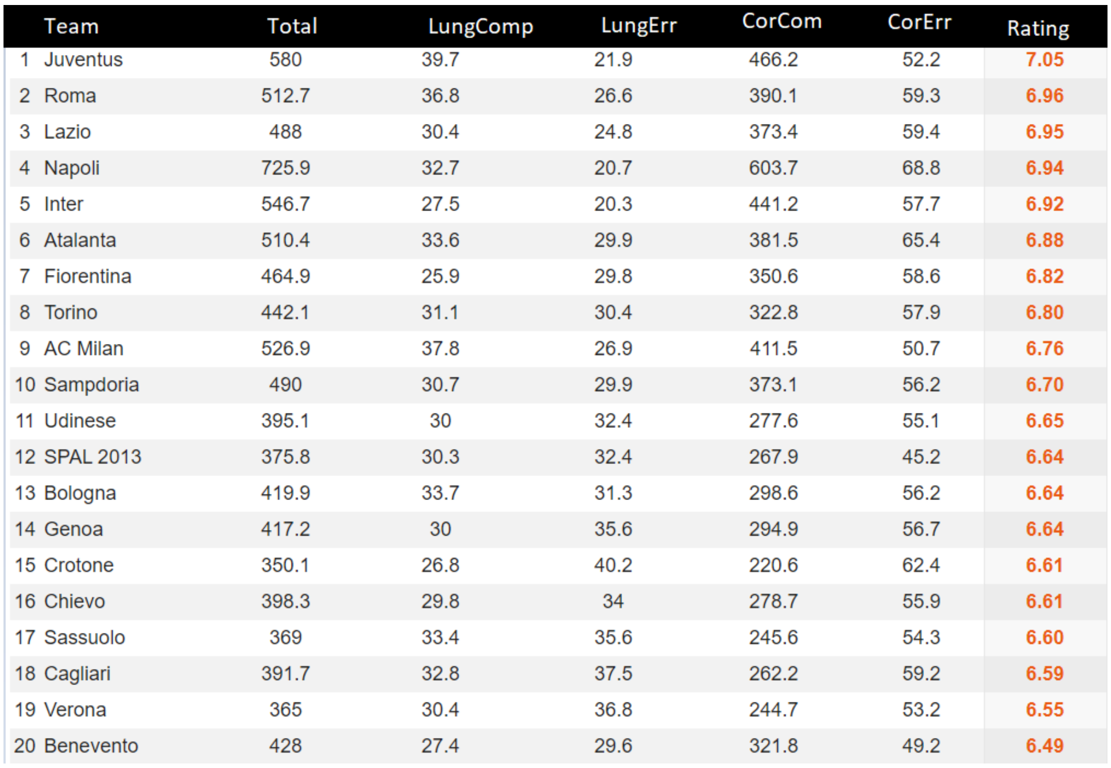
The second way to asses our methodology is the comparison using other sources of information. I have considered WhoScored[4], which data are provided by ENetPulse[5] and OptaSport[6], two of the most important sport data providers in the world.

This web site provides metrics related to the style of a soccer team. In particular, I have validated the properties Style 2, from the information extracted by the WhoScored website. I can deduce that our results are consistent to those reported by WhoScore. It is sufficient to compare our results, i.e., Table 4.8, with the WhoScore results, i.e., Figure 4.16.

Note that, with our methodology, the analysis of Style 1 and Style 3 can been seen as novelty. In fact, for the Style 1 I have defined, with the domain expert, as additional contribution, a property based on information, such as center of gravity and area that are not directly considered from soccer statistical website, since I have extracted this data through the application of image processing technique on several images refereed to several soccer matches. With regard to Style 3, WhoScored does not consider bio-mechanical information.

Our methodology offers several advantages in comparison with the above methods. The first important aspect is that a very limited number of features have be taken into account (4 features for the Styles 1 and 3, and 2 features for the Style 2), while statistical web sites usually consider a very extended number of features, for instance the WhoScore website examines over 200 raw features[7]. Thus, I obtain the same results using a small number of features, while, for example, the data providers use a process

[4]https://www.whoscored.com/
[5]https://www.enetpulse.com/
[6]https://www.optasports.com/
[7]https://it.whoscored.com/Feeds/Ratings



	Team	Total	LungComp	LungErr	CorCom	CorErr	Rating
1	Juventus	580	39.7	21.9	466.2	52.2	7.05
2	Roma	512.7	36.8	26.6	390.1	59.3	6.96
3	Lazio	488	30.4	24.8	373.4	59.4	6.95
4	Napoli	725.9	32.7	20.7	603.7	68.8	6.94
5	Inter	546.7	27.5	20.3	441.2	57.7	6.92
6	Atalanta	510.4	33.6	29.9	381.5	65.4	6.88
7	Fiorentina	464.9	25.9	29.8	350.6	58.6	6.82
8	Torino	442.1	31.1	30.4	322.8	57.9	6.80
9	AC Milan	526.9	37.8	26.9	411.5	50.7	6.76
10	Sampdoria	490	30.7	29.9	373.1	56.2	6.70
11	Udinese	395.1	30	32.4	277.6	55.1	6.65
12	SPAL 2013	375.8	30.3	32.4	267.9	45.2	6.64
13	Bologna	419.9	33.7	31.3	298.6	56.2	6.64
14	Genoa	417.2	30	35.6	294.9	56.7	6.64
15	Crotone	350.1	26.8	40.2	220.6	62.4	6.61
16	Chievo	398.3	29.8	34	278.7	55.9	6.61
17	Sassuolo	369	33.4	35.6	245.6	54.3	6.60
18	Cagliari	391.7	32.8	37.5	262.2	59.2	6.59
19	Verona	365	30.4	36.8	244.7	53.2	6.55
20	Benevento	428	27.4	29.6	321.8	49.2	6.49

Figure 4.16: *Screen obtained from WhoScored.com. (Total = Total Passes / LungComp = Long Completed Passes / LungErr = Long Wrong Passes / CorComp = Short Completed Passes / CorErr = Short Wrong Passes)*

in which data have to be recorded, analysed and distributed using a bespoke system. Moreover, to demonstrate the applicability of our method, I have verified a set of behavioural properties different from the ones proposed by soccer statistical web sites. In fact, I have formulated, as an example, the Style 1 property. Indeed, the proposed approach can be considered to automatically verify any behaviour, for this reason our approach can be considered as an ad-hoc solution customizable by the coach.

Typically, the coach provides different schemes according to the team to be faced, but also in relation to the available players for a specific match. Using the proposed approach, the coach can decide different playing styles to check for the specific match. This aspect differentiates our method with the various soccer statistics available from dedicated websites. In fact, soccer statistics websites provide generic evaluations on the teams (usually to predict odds for bets), while our proposal is related to an automatic software tool to support the coach decisions for the tactics to adopt.

Furthermore, recently many accurate decision support systems, based on machine learning, have been constructed as black boxes, that is as systems that hide their internal logic to the user. They are mainly based on machine learning models that train on massive data-sets, but with the risk to create and use decision systems that I do not really understand. Differently, the proposed method offers the explainability and the interpretability of the results. In fact, whether the formula is evaluated as false on the model by the formal verification environment, there is the possibility to perform a deep analysis to understand the reason why this happened. In this way the coach can implement (and evaluate) a different tactic (described by its ownership). Thus, the coaches can trust their results if they understand and validate the underlying rationale of their formal method based support decision system. Thus, an important difference between our method and machine learning technique is precisely that, with formal methods, I are able to explain the results and to provide a system which does not hide its internal logic to the user, represented by the coach, in this context. Below I show how a case study showing how the proposed approach can be useful to support the coach.

Let us consider the following $P1$ *CCS* process fragment, related to the first day of the season of the Benevento team.

$$\begin{aligned} proc \quad P1 = \quad & day1.(Benevento\_team.VeryLow\_F1.Low\_F2.P2+ \\ & Benevento\_team.Low\_F2.VeryLow\_F1.P2+ \\ & VeryLow\_F1.Benevento\_team.Low\_F2.P2+ \\ & VeryLow\_F1.Low\_F2.Benevento\_team.P2+ \\ & Low\_F2.Benevento\_team.VeryLow\_F1.P2+ \\ & Low\_F2.VeryLow\_F1.Benevento\_team.P2) \end{aligned}$$

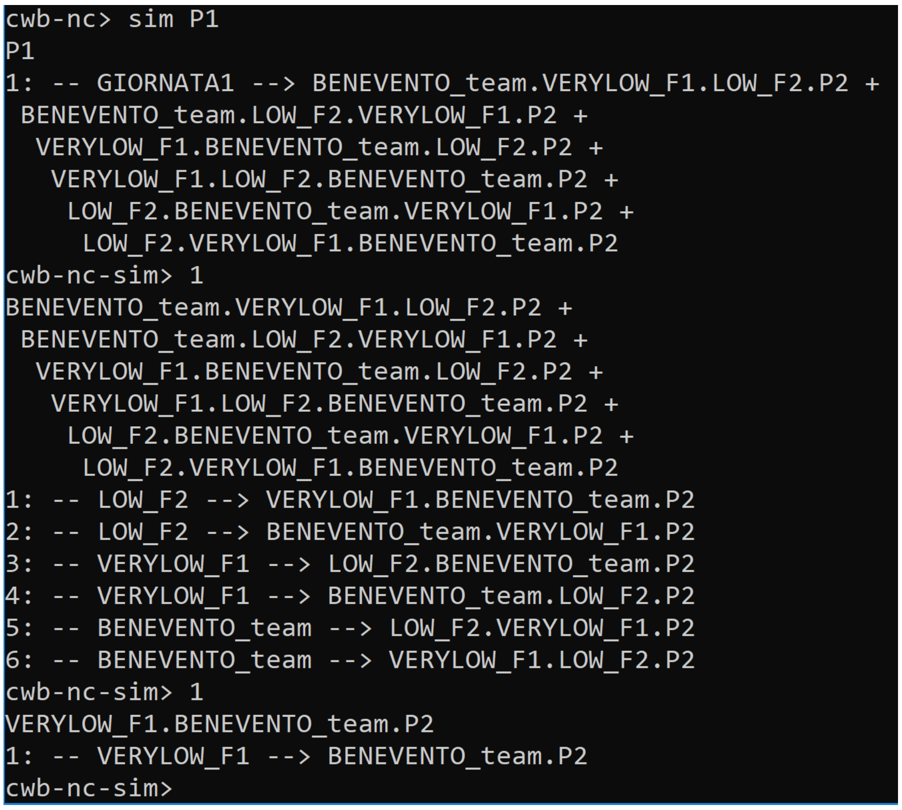
The coach wants to evaluate if the team exhibits an offensive play style in terms of ball possession, for this reason he/she evaluates the Style 2 formula. I recall that the Style 2 formula is true if the value of the two features considered in the model (i.e., overall ball possession of the team and ball possession in the opponent half of the field) are equals to *High*, *VeryHigh* and *VeryVeryHigh*.

The *Dunuen* tool outputs false, this is symptomatic that in the day1 the Benevento team exhibited a not offensive play style and this is confirmed by the features considered in the day1 model (with values equal to *Low* and *VeryLow*). In fact, as confirmation of a not offensive style, the Benevento team lost this match against the Sampdoria team (2 goals for the Sampdoria team and 1 goal for the Benevento one).

The coach through the usage of the counter-example (typical of formal methods based approached) is able to understand the reason why the day1 *CCS* model is labelled as false when the Style 2 property is checked.

Figure 4.17 shows the counter-example for the day1 *CCS* model.

Considering that the Style 2 property expects *High*, *VeryHigh* and *VeryVeryHigh* values for the analysed features, the coach from the counter-examples can understand that the team does not have an offensive style due to the presence of *Low* and *VeryLow* overall ball possession of the team and ball possession in the opponent half of the field and, consequently, can activate the tactics necessary for the change of style, for instance suggesting to the team to exhibit a more aggressive play style. To summarise, by analysing the results obtained, the coach can understand that one of the factors for which his team is losing is that it fails to have a good percentage of possession. Clearly, the more a team has the ball, the more it can make a goal. Let us analyse the second $P27$ *CCS* process fragment, shown below, related to the day27 of the Benevento team for the same season.



```
cwb-nc> sim P1
P1
1: -- GIORNATA1 --> BENEVENTO_team.VERYLOW_F1.LOW_F2.P2 +
 BENEVENTO_team.LOW_F2.VERYLOW_F1.P2 +
  VERYLOW_F1.BENEVENTO_team.LOW_F2.P2 +
   VERYLOW_F1.LOW_F2.BENEVENTO_team.P2 +
    LOW_F2.BENEVENTO_team.VERYLOW_F1.P2 +
     LOW_F2.VERYLOW_F1.BENEVENTO_team.P2
cwb-nc-sim> 1
BENEVENTO_team.VERYLOW_F1.LOW_F2.P2 +
 BENEVENTO_team.LOW_F2.VERYLOW_F1.P2 +
  VERYLOW_F1.BENEVENTO_team.LOW_F2.P2 +
   VERYLOW_F1.LOW_F2.BENEVENTO_team.P2 +
    LOW_F2.BENEVENTO_team.VERYLOW_F1.P2 +
     LOW_F2.VERYLOW_F1.BENEVENTO_team.P2
1: -- LOW_F2 --> VERYLOW_F1.BENEVENTO_team.P2
2: -- LOW_F2 --> BENEVENTO_team.VERYLOW_F1.P2
3: -- VERYLOW_F1 --> LOW_F2.BENEVENTO_team.P2
4: -- VERYLOW_F1 --> BENEVENTO_team.LOW_F2.P2
5: -- BENEVENTO_team --> LOW_F2.VERYLOW_F1.P2
6: -- BENEVENTO_team --> VERYLOW_F1.LOW_F2.P2
cwb-nc-sim> 1
VERYLOW_F1.BENEVENTO_team.P2
1: -- VERYLOW_F1 --> BENEVENTO_team.P2
cwb-nc-sim>
```

Figure 4.17: *The counter-example.*

$$
\begin{aligned}
proc \quad P27 = \; & day27.(Benevento\_team.VeryHigh\_F1.VeryHigh\_F2.P28+\\
& Benevento\_team.VeryHigh\_F2.VeryHigh\_F1.P28+\\
& VeryHigh\_F1.Benevento\_team.VeryHigh\_F2.P28+\\
& VeryHigh\_F1.VeryHigh\_F2.Benevento\_team.P28+\\
& VeryHigh\_F2.Benevento\_team.VeryHigh\_F1.P28+\\
& VeryHigh\_F2.VeryHigh\_F1.Benevento\_team.P28)
\end{aligned}
$$

As shown from the model, there are all *VeryHigh* values for the considered features: this is confirmed by the *Dunuen* tool that outputs true when it is verifying the Style 2 property. The coach understands that the team actually considered his advice and the style change has been confirmed by the *Dunuen* output. In fact, this match was won by the Benevento team with a number of goals equal to 3 (the other team, the Hellas Verona, made only 1 goal in this match). For this reason the coach can summarise that the Benevento team won the match for the changing of style, from not offensive to offensive.

CHAPTER 5

Conclusion

5.1 Conclusion

The focus of my work was about the concept of sport analytic. The research was carried on starting from three different steps: the study of human activity in the context of behavioural analysis; the application of machine learning algorithms on soccer data and the usage of formal method techniques on the same data.

About behavioural analysis, I considered two different scenarios: human activity recognition with wearable devices data and driver detection activity. In the first scenario, I proposed a method in order to recognize human activities and detect users using features gathered from accelerometer sensors, widespread in wearable and mobile devices. I exploited machine learning to build model that was able to discriminate between a set of user activities: sitting, sitting down, standing, standing up and walking. Furthermore, I demonstrated that the proposed method was able to distinguish between different users and to identify the user genre. In the second scenario, it has been defined a method for detecting the driver of a car, in real-time, exploiting supervised machine learning techniques, starting from data extracted with an OBD system. In both cases, the experimental analysis performed on real-world data shows that the proposed methods obtained encouraging results.

About the application of machine learning techniques on soccer data, I focused on the prediction of soccer match results and the analysis of soccer player positions. In the first case, I proposed a new feature set (related to the match and to players) aimed to model a soccer match. The set was related to characteristics obtainable, in real-time, when the match is in progress. I considered machine learning techniques to predict the results of the match and the number of goals, evaluating a dataset of real-world data obtained from the Italian Serie A league in the 2017-2018 season. The best results were obtained with the Random Forest algorithm, with a precision of 0.857 and a recall of

0.750 in won match prediction. While, for the goal prediction, it has been obtained a precision of 0.879 in the number of goal prediction less than two, and a precision of 0.8 in the number of goal prediction equal or greater to two.

About the prediction of soccer player positions, I proposed an approach used to recognize the player position in a soccer match, predicting the specific zone in which the player was located in a specific moment, based on predictors such as the ball position, body orientation and speed of the players involved in the analysis. It has been used supervised machine learning techniques by considering a dataset obtained through video capturing and tracking system. The data analyzed refer to several professional soccer games captured in late 2013 at the Alfheim Stadium in Tromso, Norway. The approach can be used in real-time, in order to verify if a player is playing according to the guidelines of the coach. The best results were obtained with Random Tree algorithm.

In the last step, about the application of formal methods in sport analytics, I proposed a method able to detect the style of a soccer teams, providing explainability and interpretability of the results. I modelled soccer teams in terms of automata and, by exploiting model verification techniques, I verified whether a style, expressed by means of a temporal logic formula, was exhibited by the team under analysis. The experimental analysis confirmed the effectiveness of the proposed method in soccer team behaviour detection, obtaining promising results, compared with standard baseline approaches.

As future work, it could be possible to construct a system that, analyzing data in real-time, give automatic directives to coaches and players in order to improve the performance of individual players and the entire team.

Bibliography

[1] User identification from walking activity data set. https://archive.ics.uci.edu/ml/datasets/User+Identification+From+Walking+Activity. [Online; accessed 24-October-2018].

[2] Wearable computing: Classification of body postures and movements (puc-rio) data set. http://archive.ics.uci.edu/ml/datasets/Wearable+Computing%3A+Classification+of+Body+Postures+and+Movements+%28PUC-Rio%29. [Online; accessed 24-October-2018].

[3] Zephyr bioharness bt website. https://www.zephyranywhere.com/system/components. [Online; accessed 24-October-2018].

[4] Nik Rosli Abdullah, Nafis Syabil Shahruddin, Aman Mohd Ihsan Mamat, Salmiah Kasolang, Aminuddin Zulkifli, and Rizalman Mamat. Effects of air intake pressure to the fuel economy and exhaust emissions on a small si engine. *Procedia Engineering*, 68:278–284, 2013.

[5] Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

[6] Baboota, Rahul, Kaur, and Harleen. Predictive analysis and modelling football results using machine learning approach for english premier league. *International Journal of Forecasting*, 2018.

[7] C. Baier and J. P. Katoen. *Principles of Model Checking*. 2008.

[8] Ling Bao and Stephen S Intille. Activity recognition from user-annotated acceleration data. In *International Conference on Pervasive Computing*, pages 1–17. Springer, 2004.

[9] Roger Bartlett. Performance analysis: can bringing together biomechanics and notational analysis benefit coaches? *International Journal of Performance Analysis in Sport*, 1(1):122–126, 2001.

[10] Martin Berchtold, Matthias Budde, Dawud Gordon, Hedda R Schmidtke, and Michael Beigl. Actiserv: Activity recognition service for mobile phones. In *Wearable Computers (ISWC), 2010 International Symposium on*, pages 1–8. IEEE, 2010.

[11] Martin Berchtold, Matthias Budde, Hedda R Schmidtke, and Michael Beigl. An extensible modular recognition concept that makes activity recognition practical. In *Annual Conference on Artificial Intelligence*, pages 400–409. Springer, 2010.

[12] Lopes Berrar and Dubitsky. Incorporating domain knowledge in machine learning for soccer outcome prediction. *Special Issue on Machine Learning for Soccer*, 2019.

[13] Jonathan Bowen. Formal methods in safety-critical standards. In *Proceedings 1993 Software Engineering Standards Symposium*, pages 168–177. IEEE, 1993.

[14] Gerardo Canfora, Fabio Martinelli, Francesco Mercaldo, Vittoria Nardone, Antonella Santone, and Corrado Aaron Visaggio. Leila: formal tool for identifying mobile malicious behaviour. *IEEE Transactions on Software Engineering*, 2018.

[15] Gerardo Canfora, Francesco Mercaldo, Giovanni Moriano, and Corrado Aaron Visaggio. Composition-malware: building android malware at run time. In *Availability, Reliability and Security (ARES), 2015 10th International Conference on*, pages 318–326. IEEE, 2015.

[16] Capobianco, Di Giacomo, Martinelli, Mercaldo, and Santone. Wearable devices for human activity recognition and user detection. In *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 365–372. IEEE, 2019.

[17] Capobianco, Di Giacomo, Mercaldo, and Santone. A formal methodology for notational analysis and real-time decision support in sport environment. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5305–5307. IEEE, 2018.

[18] Capobianco, Di Giacomo, Mercaldo, and Santone. Dunuen: A user-friendly formal verification tool. Procedia Computer Science, 2019.

[19] Giovanni Capobianco, Umberto Di Giacomo, Francesco Mercaldo, Vittoria Nardone, and Antonella Santone. Can machine learning predict soccer match results? In *11th International Conference on Agents and Artificial Intelligence*, pages 458–465, 01 2019.

[20] Jaime G Carbonell, Ryszard S Michalski, and Tom M Mitchell. An overview of machine learning. In *Machine Learning, Volume I*, pages 3–23. Elsevier, 1983.

[21] Carmeli, Abraham, Schaubroeck, and John. Top management team behavioral integration, decision quality, and organizational decline. *The Leadership Quarterly*, 17(5):441–453, 2006.

[22] Pierluigi Casale, Oriol Pujol, and Petia Radeva. Personalization and user verification in wearable systems using biometric walking patterns. *Personal and Ubiquitous Computing*, 16(5):563–580, 2012.

[23] Sangjo Choi, Jeonghee Kim, Donggu Kwak, Pongtep Angkititrakul, and John Hansen. Analysis and classification of driver behavior using in-vehicle can-bus information. *Bienn. Workshop DSP In-Veh. Mob. Syst.*, 01 2007.

[24] Cimatti, Clarke, and Roveri. Nusmv: A new symbolic model verifier. volume 2, pages 410–425, 2000.

[25] Aniello Cimitile, Fabio Martinelli, Francesco Mercaldo, Vittoria Nardone, and Antonella Santone. Formal methods meet mobile code obfuscation identification of code reordering technique. In *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 263–268. IEEE, 2017.

[26] Aniello Cimitile, Francesco Mercaldo, Fabio Martinelli, Vittoria Nardone, Antonella Santone, and Gigliola Vaglini. Model checking for mobile android malware evolution. In *Proceedings of the 5th International FME Workshop on Formal Methods in Software Engineering*, pages 24–30. IEEE Press, 2017.

[27] Aniello Cimitile, Francesco Mercaldo, Vittoria Nardone, Antonella Santone, and Corrado Aaron Visaggio. Talos: no more ransomware victims with formal methods. *International Journal of Information Security*, 17(6):719–738, 2018.

[28] Clemente, Filipe, Couceiro, Micael, Martins, and Fernando. Soccer teams behaviors: analysis of the team's distribution in function to ball possession. *Res. J. Appl. Sci. Eng. Technol*, 6(1):130–136, 2013.

[29] J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30, Jan 2004.

[30] Carlos Dietrich, David Koop, Huy T Vo, and Cláudio T Silva. Baseball4d: A tool for baseball game reconstruction & visualization. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 23–32. IEEE, 2014.

[31] Ap Dijksterhuis, Maarten W Bos, Andries Van der Leij, and Rick B Van Baaren. Predicting soccer matches after unconscious and conscious thought as a function of expertise. *Psychological Science*, 20(11):1381–1387, 2009.

[32] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018.

[33] Miro Enev, Alex Takakuwa, Karl Koscher, and Tadayoshi Kohno. Automobile driver fingerprinting. *Proceedings on Privacy Enhancing Technologies*, 2016(1):34–50, 2016.

[34] Bornn Fernandez and Cervone. Decomposing the immeasurable sport: A deep learning expected possession value framework for soccer. *Sport Analytics Conference*, 2019.

[35] R. W. Frischholz and U. Dieckmann. Biold: a multimodal biometric identification system. *Computer*, 33(2):64–68, Feb 2000.

[36] Hubert Garavel, Frédéric Lang, Radu Mateescu, and Wendelin Serwe. CADP 2011: a toolbox for the construction and analysis of distributed processes. *STTT*, 15(2):89–107, 2013.

[37] Susan Gerhart, Dan Craigen, and Ted Ralston. Experience with formal methods in critical systems. *IEEE Software*, 11(1):21–28, 1994.

[38] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[39] Gomes, João, Portela, Filipe, Santos, and Manuel Filipe. Decision support system for predicting football game result. In *Computers-19th International Conference on Circuits, Systems, Communications and Computers-Intelligent Systems and Applications Special Sessions. Series*, volume 32, pages 348–353, 2015.

[40] S. Gonzalez, C. M. Travieso, J. B. Alonso, and M. A. Ferrer. Automatic biometric identification system by hand geometry. In *IEEE 37th Annual 2003 International Carnahan Conference onSecurity Technology, 2003. Proceedings.*, pages 281–284, Oct 2003.

[41] Constance Heitmeyer. On the need for practical formal methods. In *International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems*, pages 18–26. Springer, 1998.

[42] C Michael Holloway. Why engineers should consider formal methods. In *16th DASC. AIAA/IEEE Digital Avionics Systems Conference. Reflections to the Future. Proceedings*, volume 1, pages 1–3. IEEE, 1997.

[43] Mike Hughes and Ian Franks. Notational analysis of sport. *Journal of Sports Science and Medicine*, 3, 06 2004.

[44] K. Igarashi, C. Miyajima, K. Itou, K. Takeda, F. Itakura, and H. Abut. Biometric identification using driving behavioral signals. In *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, volume 1, pages 65–68 Vol.1, June 2004.

[45] Iglesias, José Antonio, Ledezma, Agapito, Sanchis, Araceli, and Kaminka. Classifying efficiently the behavior of a soccer team. *IAS-10*, pages 316–323, 2008.

[46] M. Itoh, D. Yokoyama, M. Toyoda, and M. Kitsuregawa. Visual interface for exploring caution spots from vehicle recorder big data. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 776–784, Oct 2015.

[47] Sushma Jain and Harmandeep Kaur. Machine learning approaches to predict basketball game outcome. pages 1–7, 09 2017.

[48] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[49] Joseph, Anito, Fenton, Norman, Neil, and Martin. Predicting football results using bayesian nets and other machine learning techniques. *Knowledge-Based Systems*, 19(7):544–553, 2006.

[50] Tzu-Ping Kao, Che-Wei Lin, and Jeen-Shing Wang. Development of a portable activity detector for daily activity recognition. In *Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on*, pages 115–120. IEEE, 2009.

[51] Ioannis Kavakiotis, Olga Tsave, Athanasios Salifoglou, Nicos Maglaveras, Ioannis Vlahavas, and Ioanna Chouvarda. Machine learning and data mining methods in diabetes research. *Computational and structural biotechnology journal*, 15:104–116, 2017.

[52] Tarak Kharrat, Javier Peña, and I.G. Mchale. Plus-minus player ratings for soccer. *European Journal of Operational Research*, 2017.

[53] John C Knight, Colleen L DeJong, Matthew S Gibble, and Luis G Nakano. Why are formal methods not used more widely? In *Fourth NASA formal methods workshop*. Citeseer, 1997.

[54] Gunjan Kumar. Machine learning for soccer analytics. 2013.

[55] Kwiatkowska, Norman, and Parker. Prism: Probabilistic symbolic model checker. volume 2324, pages 200–204, 2002.

[56] Oscar D Lara and Miguel A Labrador. A mobile platform for real-time human activity recognition. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pages 667–671. IEEE, 2012.

[57] Liti, Piccialli, and Sciandrone. Predicting soccer match outcome using machine learning algorithms. In *Proceedings of MathSport International 2017 Conference*, page 229, 2017.

[58] Guiliang Liu and Oliver Schulte. Deep reinforcement learning in ice hockey for context-aware player evaluation. pages 3442–3448, 07 2018.

[59] Steve Lohr. The age of big data. *New York Times*, 11(2012), 2012.

[60] Patrick Lucey, Alina Bialkowski, Mathew Monfort, Peter Carr, and Iain Matthews. quality vs quantity: Improved shot prediction in soccer using strategic features from spatiotemporal data. In *Proc. 8th annual mit sloan sports analytics conference*, pages 1–9, 2014.

[61] Fabio Martinelli, Francesco Mercaldo, Vittoria Nardone, Antonella Santone, Arun Kumar Sangaiah, and Aniello Cimitile. Evaluating model checking for cyber threats code obfuscation identification. *Journal of Parallel and Distributed Computing*, 119:203–218, 2018.

[62] Fabio Martinelli, Francesco Mercaldo, Albina Orlando, Vittoria Nardone, Antonella Santone, and Arun Kumar Sangaiah. Human behavior characterization for driving style recognition in vehicle system. *Computers & Electrical Engineering*, 2018.

[63] Francesco Mercaldo, Vittoria Nardone, and Antonella Santone. Diabetes mellitus affected patients classification and diagnosis through machine learning techniques. *Procedia Computer Science*, 112(C):2519–2528, 2017.

[64] Ryszard S Michalski, Jaime G Carbonell, and Tom M Mitchell. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[65] Tom M Mitchell. Machine learning and data mining. *Communications of the ACM*, 42(11):30–36, 1999.

[66] Chiyomi Miyajima, Yoshihiro Nishiwaki, Koji Ozawa, Toshihiro Wakita, Katsunobu Itou, and Kazuya Takeda. Cepstral analysis of driving behavioral signals for driver identification. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, pages V–V. IEEE, 2006.

[67] Mondello, Michael, Kamke, and Christopher. The introduction and application of sports analytics in professional sport organizations. *Journal of Applied Sport Management*, 6(2), 2014.

[68] Jean-Francois Monin. *Understanding formal methods*. Springer Science & Business Media, 2012.

[69] Nasser M Nasrabadi. Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901, 2007.

[70] Neuman, Yair, Israeli, Navot, Vilenchik, Dan, Cohen, and Yochai. The adaptive behavior of a soccer team: An entropy-based analysis. *Entropy*, 20(10):758, 2018.

[71] Norzaliza Md Nor and Abdul Wahab. Driver identification and driver's emotion verification using kde and mlp neural networks. In *Information and Communication Technology for the Muslim World (ICT4M), 2010 International Conference on*, pages E96–E101. IEEE, 2010.

[72] Henry Friday Nweke, Ying Wah Teh, Uzoma Rita Alo, and Ghulam Mujtaba. Analysis of multi-sensor fusion for mobile and wearable sensor based human activity recognition. In *Proceedings of the International Conference on Data Processing and Applications*, pages 22–26. ACM, 2018.

[73] M. Okamoto, S. Otani, Y. Kaitani, and K. Uchida. Identification of driver operations with extraction of driving primitives. In *2011 IEEE International Conference on Control Applications (CCA)*, pages 338–344, Sept 2011.

[74] David Lorge Parnas. The real risks of artificial intelligence. *Communications of the ACM*, 60(10):27–31, 2017.

[75] Svein Arne Pettersen, Dag Johansen, Håvard Johansen, Vegard Berg-Johansen, Vamsidhar Reddy Gaddam, Asgeir Mortensen, Ragnar Langseth, Carsten Griwodz, Håkon Kvale Stensland, and Pål Halvorsen. Soccer video and player position dataset. In *Proceedings of the 5th ACM Multimedia Systems Conference*, MMSys '14, page 18–23, New York, NY, USA, 2014. Association for Computing Machinery.

[76] Razali, Mustapha, Yatim, and Ab Aziz. Predicting football matches results using bayesian networks for english premier league (epl). *International Research and Innovation Summit*, 2017.

[77] Robert Rein and Daniel Memmert. Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science. *SpringerPlus*, 5(1):1410, 2016.

[78] D. A. Reynolds. An overview of automatic speaker recognition technology. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages IV–4072–IV–4075, May 2002.

[79] S. Ribaric, D. Ribaric, and N. Pavesic. Multimodal biometric user-identification system for network-based applications. *IEE Proceedings - Vision, Image and Signal Processing*, 150(6):409–416, Dec 2003.

[80] Daniele Riboni and Claudio Bettini. Cosar: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 2011.

[81] Robin Milner. *Communication and concurrency*. PHI Series in computer science. Prentice Hall, 1989.

[82] Hossein Saiedian. An invitation to formal methods. *Computer*, 29(4):16–17, 1996.

[83] Antonella Santone. Automatic verification of concurrent systems using a formula-based compositional approach. *Acta Informatica*, 38(8):531–564, 2002.

[84] Antonella Santone and Gigliola Vaglini. Abstract reduction in directed model checking ccs processes. *Acta informatica*, 49(5):313–341, 2012.

[85] Sathe, Kasat, Kulkarni, and Satao. Predictive analysis of premier league using machine learning. *International Journal of Innovative Research in Computer and Communication Engineering*, 2017.

[86] Steven Schultze and Christian-Mathias Wellbrock. A weighted plus/minus metric for individual soccer player performance. *Journal of Sports Analytics*, 4:1–11, 10 2017.

[87] Zuhaib Ahmed Shaikh, Umair Ali Khan, Muhammad Awais Rajput, and Abdul Wahid Memon. Machine learning based number plate detection and recognition. In *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods*, pages 327–333. SCITEPRESS-Science and Technology Publications, Lda, 2016.

[88] Diogo Silva, Helena Aidos, and Ana LN Fred. Efficient evidence accumulation clustering for large datasets. In *ICPRAM*, pages 367–374, 2016.

[89] Thitaree Tanprasert, Chalermpol Saiprasert, and Suttipong Thajchayapong. Combining unsupervised anomaly detection and neural networks for driver identification. *Journal of Advanced Transportation*, 2017:1–13, 10 2017.

[90] Emmanuel Munguia Tapia, Stephen S Intille, William Haskell, Kent Larson, Julie Wright, Abby King, and Robert Friedman. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart monitor. In *In: Proc. Int. Symp. on Wearable Comp*. Citeseer, 2007.

[91] Wallace Ugulino, Débora Cardador, Katia Vega, Eduardo Velloso, Ruy Milidiú, and Hugo Fuks. Wearable computing: Accelerometers data classification of body postures and movements. In *Advances in Artificial Intelligence-SBIA 2012*, pages 52–61. Springer, 2012.

[92] Shan Ullah and Deok-Hwan Kim. Lightweight driver behavior identification model with sparse learning on in-vehicle can-bus sensor data. *Sensors*, 20(18):5030, 2020.

[93] Vilar, Luís, Araújo, Duarte, Davids, Keith, Bar-Yam, and Yaneer. Science of winning soccer: Emergent pattern-forming dynamics in association football. *Journal of systems science and complexity*, 26(1):73–84, 2013.

[94] Philipp Voigt, Matthias Budde, Erik Pescara, Manato Fujimoto, Keiichi Yasumoto, and Michael Beigl. Feasibility of human activity recognition using wearable depth cameras. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, pages 92–95. ACM, 2018.

[95] A. Wahab, C. Quek, C. K. Tan, and K. Takeda. Driving profile modeling and recognition based on soft computing approach. *IEEE Transactions on Neural Networks*, 20(4):563–582, April 2009.

[96] Jeannette M Wing. A specifier's introduction to formal methods. *Computer*, 23(9):8–22, 1990.

[97] Jim Woodcock, Peter Gorm Larsen, Juan Bicarregui, and John Fitzgerald. Formal methods: Practice and experience. *ACM computing surveys (CSUR)*, 41(4):19, 2009.

[98] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

[99] Xingjian Zhang, Xiaohua Zhao, and Jian Rong. A study of individual characteristics of driving behavior based on hidden markov model. *19th Intelligent Transport Systems World Congress, ITS 2012*, 167:AP–00306, 01 2012.