Høgskolen i Østfold

Predicting Injuries in Norwegian Women's Soccer Players: A Machine Learning Approach

Master Thesis

Md Mohaiminul Islam Emon

Department of Computer Science and Communication
Østfold University College
Halden, Norway
June 15, 2024

Predicting Injuries in Norwegian Women's Soccer Players: A Machine Learning Approach

Master Thesis

Md Mohaiminul Islam Emon

Department of Computer Science and Communication
Østfold University College
Halden, Norway
June 15, 2024

Abstract

This thesis investigates the feasibility and effectiveness of machine learning in predicting injuries among Norwegian women's soccer players. We have identified certain features, including illness, weekly load, Chronic Training Load over the past 28 days, Acute Training Load, Chronic Training Load over the past 42 days, monotony, strain, and Acute:Chronic Workload Ratio, have a strong correlation with injury occurrence. We also identified the most effective time frame for predicting injuries, while algorithms including Decision Tree, K-Nearest Neighbors, LSTM, and XGBoost performed best using a 32-day time frame. Furthermore, we have investigated eight machine learning algorithms performance to predict Norwegian women's soccer players injuries. XGBoost performed the best with an F1 score of 0.58. The model achieved a recall of 0.50 and a precision of 0.71, showcasing its strong performance in predicting soccer player injuries while also minimizing the occurrence of false positive predictions.

We have conducted hyperparameter tuning on seven machine learning algorithms to evaluate their performance before and after the tuning process. After conducting hyperparameter tuning, we observed enhanced performance in algorithms such as Logistic Regression, Random Forest, K-Nearest Neighbors, Support Vector Machine, and Naive Bayes. Furthermore, we evaluate the performance of machine learning algorithms on a single team dataset. We found that K-Nearest Neighbors performed exceptionally well in predicting injuries for a single team. On the other hand, XGBoost emerged as the top performer for predicting injuries across multiple teams. We also analyzed our algorithms performance, specifically on the injured players dataset. We identified that XGBoost performed best among other algorithms, whereas Decision Tree and K-Nearest Neighbors showed moderate performance. Lastly, we have evaluated algorithms performance based on dataset size. The full dataset had a class imbalance (56 injury records, 8526 non-injured records), making it difficult to predict injuries. The single team dataset (Team A) provided better predictions within that specific context but may not be applicable to other teams. Analyzing only the dataset of injured players revealed specific patterns but may not capture broader injury trends.

Our research introduces a novel approach to predicting injuries in women's soccer players using machine learning algorithms. Furthermore, this thesis offers meaningful insights for players, coaches, medical professionals, and physicians interested in studying soccer player injuries and related factors.

Keywords: Machine learning, injury prediction, women's soccer players, SoccerMon, algorithms, hyperparameter tuning, performance metrics.

Acknowledgments

I would like to thank my internal supervisor, Accociate Professor Lars Vidar Magnusson, from Østfold University College, Halden, Norway. His guidance and unwavering support have been invaluable throughout my research journey. He consistently encouraged me to step out of my comfort zone, explore new possibilities, and surpass my limitations. I am truly grateful for his mentorship and dedication.

Moreover, I would like to extend my sincere appreciation to my external supervisors, Chief Research Scientist Pål Halvorsen, Research Professor Michael Riegler and Postdoctoral Fellow Cise Midoglu from Simula Metropolitan Center for Digital Engineering, Oslo, Norway. Their unwavering support, guidance, and encouragement have been instrumental in shaping my research journey. I am grateful for their valuable feedback, insightful ideas, and expert advice, which have greatly contributed to the quality of my thesis. Regular meetings with them to discuss my findings and future directions have been immensely helpful in crafting a strong and impactful thesis.

I also want to thank my family for their constant love and encouragement that has helped me stay motivated throughout my degree.

Finally, I would like to thank my dear Nasrin Chowdhury for all the support and for always turning tough times into good times.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Soccer is one of the most famous games in the world, with a huge fanbase. According to Wikipedia, around 250 million active soccer players in 200 countries [15][6]. The soccer market is enormous and continues to expand annually. According to a Deloitte report, the European market expanded by 7% to reach 29.5 billion euros during the 2021/22 session. Other soccer leagues like the La Liga, Bundesliga, and Premier League are growing quickly in terms of review, club size, and fan support. At the start of each season, clubs seek talented and fit players to join their teams. They invest a lot of money in buying new players. In modern times, teams actively keep track of their players health, performance, and injury history. This information is used to analyze player capabilities, prevent injuries, and provide valuable insights to coaches and team management about the overall team situation. This study focuses on using machine learning techniques to understand and prevent injuries in women's soccer.

The main objective of this thesis is to investigate the features contributing to injuries and predict injury risk using machine learning algorithms. To achieve this, we have divided our research objective into seven sub-questions. Previous studies have primarily focused on subjective or objective player metrics, such as wellness, training load, and game performance, to predict injuries [55] [50], readiness [48], and game performance [58]. However, our approach takes a different path by integrating both subjective and objective GPS data from SoccerMon. This unique approach will enable us to provide more accurate and insightful answers to our research objectives and sub-research questions.

1.1 Motivation

Besides men's soccer, the growth of women's soccer has been remarkable over the past few decades, with increasing participation, visibility, and recognition across the world. More and more women are participating in soccer at various levels. Behind the growth of women's soccer, leagues such as the National Women's Soccer League (NWSL) in the United States, the FA Women's Super League in England, and Division 1 Féminine in France played key roles [49]. Many investors and large corporations are now providing financial support for women's soccer. Greater media coverage and broadcasting of women's soccer matches have significantly increased the visibility of the sport. This exposure helps in attracting fans, sponsors, and support for women's soccer. There are 211 members of FIFA, and almost all the member countries have women's soccer teams [23]. The women's national teams participate in various international competitions, including regional tournaments like the UEFA Women's Championship, CONCACAF Women's Championship, AFC

Women's Asian Cup, CAF Women's Africa Cup of Nations, OFC Women's Nations Cup, and others. The value of the women's soccer market has been steadily increasing, reflecting the growth and rising popularity of women's soccer globally. According to a recent UEFA report, women's soccer in Europe may see a six-fold growth in its economic value over the next ten years, reaching over £578 million annually.

Soccer is a highly dynamic and fast-paced sport that demands agility, skill, strategy, and teamwork. Maintaining optimal health is critical to winning the game and is a highly desired quality in each member of a team. The team physician and coach have responsibilities for making sure a player is in optimal physical condition and healthy. One of the primary concerns for coaches and players alike is the possibility of injury during hard training sessions or in competitive play. Preventing injuries stands as a top priority for both players and coaching staff. A player who is in optimal physical shape can make a significant positive impact on their team's ability to secure victories, whereas an injured player has the potential to hinder overall team performance. The player's injury cost over £500 million in the 2021/22 season, according to the findings of the European Football Injury Index [54]. In addition, the recovery period for injured players is frequently lengthy, which adversely impacts team spirit, finances, and other important factors. In recent years, there has been a growing focus on reducing the risk of injuries in soccer [8]. Many clubs and research organizations have done several research projects, and there are several ongoing projects to investigate the reasons behind the players injuries as well as to find out injury prevention strategies. Experts in the industry recommend several strategies for injury prevention. Proper warm-up exercises before the game are crucial to prepare players and minimize the risk of injury. Balance and stability exercises are also important to enhance performance and prevent joint injuries. Adequate rest and recovery after training sessions and games are essential for players. Additionally, maintaining proper hydration and a balanced diet supports overall health, muscle recovery, and injury prevention. It is also recommended to promptly address minor injuries and seek professional medical advice to prevent them from worsening.

There are various factors that can contribute to a soccer player's injury. These factors may include intense training sessions preceding a competitive match, illness, mood, or other wellness-related elements. Nowadays, soccer clubs are more interested in collecting players data and using the data for further analysis. subjective metrics, such as player wellness reports, training load, injury records, illness history, and game performance parameters, are commonly used for analyzing and predicting injury, player performance, etc. The author, Anna Linnea Jarmann [55], used SoccerMon [52] subjective metrics to identify the injury risk factors for elite soccer teams using survival analysis. There are also several existing studies where researchers used objective metrics to analyze and predict participants' readiness, injury, game performance, and so on. For example, the author, Lars Hoel [50], used the GPS parameters from SoccerMon objective metrics to visualize and predict features. In our thesis, we used both subjective and objective metrics to predict the injuries of women's soccer players using machine learning algorithms. We have extracted the GPS features from objective metrics and combined them with subject data such as wellness, training load, illness, injury, and game performance parameters to predict injury.

One of the most promising approaches for injury prediction is to use machine learning techniques. Machine learning can help prevent and reduce the risk of injury to soccer players [28] [50]. By using machine learning techniques, it is possible to find the patterns and trends of soccer players that might be more useful for players, coaches, and team management officials. This kind of approach

helps players understand their condition after a game or training session, coaches can get better insights, and team management can make decisions based on the report. While there is existing research on the topic, there is a noticeable gap when it comes to evaluating the performance of machine learning algorithms in identifying injury risk using both subjective and objective metrics. We recognized the need for further exploration in this area and sought to find answers. In addition, our thesis addresses specific research questions and provides results and explanations to enhance understanding.

1.2 Research Questions

Our study aims to investigate various types of machine learning algorithms for identifying injury risk in women's soccer players. The primary research objective of this thesis is:

How can state-of-the-art machine learning algorithms predict and help to reduce the risk of injuries in women's soccer?

To address our main research objective, we have divided it into seven sub-questions. The SoccerMon [52] dataset provides various features, both subjective and objective. However, we are uncertain about the importance of these features in predicting injuries. Thus, our investigation focuses on identifying the most relevant features that are related to injuries. Additionally, we have investigated the optimal time frame for predicting injuries, as we are unsure about the ideal duration for predicting injuries in women's soccer players for the following day. Moreover, we are uncertain about the best machine learning algorithm for injury prediction. Hence, we have explored eight machine learning algorithms to determine the most effective one for predicting injuries in women's soccer players.

We are unsure about the performance of our machine learning algorithms after hyperparameter tuning. Therefore, our thesis investigates their performance both before and after tuning. Moreover, we do not have an idea if our machine learning algorithms will perform the same for all the teams. Therefore, we investigate how the algorithms perform on multiple teams and on a single team. Additionally, what if a soccer club only wants to focus on previously injured players and analyze their data to identify the risk of injury for those specific players? We do not know how our machine learning algorithms will perform on the dataset of only injured players. So, we investigate our algorithms' performance to predict injury using only the injured players dataset. Moreover, we are uncertain about how our machine learning model will perform on different datasets. To address this uncertainty, we evaluate the performance of the algorithms on datasets of varying scales. This investigation helps us understand how effectively the algorithms can adapt to and perform across different sizes of datasets.

RQ1: What are the most important features that are correlated with injuries in women's soccer players?

RQ2: How many days is the most effective time frame for predicting injuries?

RQ3: Which machine learning algorithm is most effective for predicting injuries for the following day?

RQ4: How does the hyperparameter tuning have an influence on improving the performance of the algoritms?

RQ5: Is it more effective to use a single algorithm for all teams or to develop separate algorithms for each team when predicting injuries?

RQ6: How do machine learning algorithms perform when evaluated on a dataset containing only injured players?

RQ7: How does the scale of the dataset have an impact on the performance of machine learning algorithms?

1.3 Scope

This thesis aims to use machine learning algorithms to predict injuries among Norwegian women's soccer players. We have used the SoccerMon [52] dataset for our thesis, which consists of data collected from two top Norwegian soccer teams during 2020 and 2021. This dataset includes both subjective and objective metrics about the soccer players. We have used both these subjective metrics and GPS features extracted from the objective metrics for our study. The subjective metrics include player wellness, training load, injury, illness, and game performance parameters. Our analysis has identified the key factors that are highly correlated with injuries. Additionally, we have also identified the features that have moderate and weak correlations with injuries. This information can be valuable for developing effective training and recovery programs for soccer players. Furthermore, we have conducted research to determine the optimal time frame for predicting injuries. The findings from this experiment can be beneficial for future researchers in this field, as they can utilize this knowledge to identify the most suitable time frame for predicting injuries in their studies.

In our study, we have used eight different machine learning algorithms to predict injuries. In the results and discussion chapter, we have provided a brief overview of the performance and usefulness of these algorithms. This information can be used as a basis for future research. Furthermore, we have evaluated the performance of our machine learning algorithms on a specific team (Team A) dataset to evaluate their effectiveness on a smaller scale. These findings can help researchers determine whether it is more effective to develop a common machine learning framework for multiple teams or to create separate frameworks for each team when predicting injuries. Additionally, we have considered the performance of our algorithms specifically for injured players, as coaches and team management often prioritize their recovery. Lastly, we investigate the impact of dataset size on the performance of our machine learning algorithms, providing guidance for future experiments in selecting appropriate dataset sizes.

The findings of this thesis make significant contributions to both machine learning and soccer. We demonstrate how to handle an imbalanced dataset, evaluate the performance of different machine learning algorithms, explore the effects of hyperparameter tuning, and more. We believe our findings will help reduce injury risks for players. Additionally, our work will be valuable to other researchers focusing on the health and injuries of soccer players.

1.4 Ethical Considerations

In this thesis, we aim to predict injuries among women's soccer players using machine learning techniques. Our analysis is based on the SoccerMon [52] dataset, which includes information on players from two Norwegian soccer teams. Since this data contains both subjective and objective details, including personal and sensitive information, it is crucial to prioritize the privacy and security of the participants. To achieve this, we have taken steps to anonymize the data thoroughly. This involves removing all metadata and using randomly generated file names, a widely accepted method for protecting data privacy. Additionally, all players are fully informed about how the data is collected, its nature, and the purpose of its use. This ensures transparency and ethical handling of the data throughout the research process.

To complete this thesis, we used various internal resources that are not publicly accessible. This confidentiality is critical to protecting the personal information of the teams and players being studied. Despite the limitations on data accessibility, the insights gained from this private information are significant and should be included in the thesis. While the findings may not be comparable due to the private nature of the data, they nonetheless provide value to the research.

1.5 Main Contributions

The primary objective of this thesis is to develop a machine learning framework for predicting the risk of injury among Norwegian elite women's soccer players. The ultimate goal is to assist elite soccer teams in reducing the risk of injury. Additionally, this research aims to help players, coaches, and team management gain a better understanding of the main factors linked to injuries. By identifying these factors, the research can offer valuable insights into injury prevention and player management strategies.

Several previous studies have used the SoccerMon dataset to predict player readiness, game performance, injuries, and so on. For instance, Anna Linnea Jarmann proposed a survival analysis technique to identify injury risk factors, focusing solely on subjective metrics. In a similar vein, Lars Hoel used athlete GPS monitoring data to visualize and predict various features. In our thesis, we have identified a gap in the existing research, as previous studies have primarily focused on either subjective or objective metrics for injury prediction. To address this gap, we have integrated both subjective metrics and objective GPS features into our analysis.

- At the early stage, we have proposed a pipeline to run the experiment to predict the risk of injury for elite Norwegian women's soccer players. The pipeline contains data preprocessing, exploratory data analysis (EDA), feature engineering, model training, injury prediction, and model evaluation.
- We have provided detailed information about the SoccerMon dataset through exploratory data analysis. The EDA part includes correlations between features, outliers of various numerical features, and time series analyses to identify the correlation levels of various performance and well-being metrics.
- In this thesis, we have discovered and highlighted potential causes of injury in women's soccer players. Features such as illness, weekly_load, ctl28, atl, ctl42, monotony, strain, and acwr have a great influence on injury occurrence. There are some other features, such

as daily_load, fatigue, average_running_speed, soreness, and top_speed have moderately correlation with injury. This information can be used to create effective strategies to prevent injuries.

- We have used the window function to identify the most effective time frame for injury prediction. We have used the window sizes of 2, 4, 8, 16, and 32. Based on our experiment, we have found that the best time frame for injury prediction is 32 days.
- We have used eight machine learning algorithms, including Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, LSTM, Support Vector Machine, XGBoost, and Naive Bayes, and evaluated their performance. Among all the algorithms, XGBoost performed best with an F1 score of 0.58 and recall and precision scores of 0.50 and 0.71, respectively.
- At a later stage, we have investigated the influence of hyperparameter tuning on the performance of the machine learning algorithms. Our findings demonstrate how fine-tuning hyperparameters significantly improve algorithm accuracy, precision, and recall, thereby enhancing the reliability of injury predictions.
- Through our experiments, we have explored whether multiple teams can use a common machine learning framework to predict injuries or if each team benefits from a separate framework. This analysis provides valuable insights into how well the developed algorithms can be applied in different team settings.
- We have conducted experiments to evaluate the performance of the algorithms on datasets containing only injured players, providing insights into the algorithms sensitivity to injury patterns.
- Additionally, We also explored the impact of dataset scale on algorithms performance and highlighted the importance of data quantity and quality in injury prediction.

The novelty of this thesis is that it includes the prediction of injury for two Norwegian elite soccer teams using different types of machine learning algorithms using both subjective metrics and GPS features from objective metrics. Moreover, the dataset had mentionable missing values, and we introduced a random forest imputer to handle the missing values. Furthermore, we have introduced oversampling and undersampling techniques to handle an imbalanced dataset. In summary, our approach gives valuable insights to predict the risk of injury.

1.6 Thesis Outline

The remaining content of the thesis is structured into the following six chapters.

- **Chapter 2: Background** explored the PmSys framework with a focus on mobile apps, web apps, and soccer dashboards. We also presented an overview of the SoccerMon dataset that we used for injury prediction. We introduced basic machine learning and explained the algorithms used in our thesis for injury prediction. Additionally, we discussed different sampling techniques to handle imbalanced data. Furthermore, we discussed various metrics for monitoring athlete health and performance, including training, wellness, illness, and game performance. We also examined the implementation of AI for injury prediction in soccer and reviewed existing research in this area.

- **Chapter 3: Methodology and Implementation** one of the most crucial chapters within this thesis is the methodology and implementation, where we provide in-depth in-depth knowledge to build the machine learning framework to predict the injuries of Norwegian women's soccer players. This chapter serves as a comprehensive guide, covering a range of essential aspects, including the proposed pipeline, import dataset, data preprocessing, exploratory data analysis, feature engineering, final datasets, and injury prediction. By meticulously detailing each of these components, we aim to present a comprehensive understanding of the practical execution of our thesis.
- **Chapter 4: Experiments and Results** presents a thorough analysis of the algorithm's performance and evaluation. Through a series of seven experiments, we provide comprehensive results and findings for each of these experiments. Furthermore, we include preliminary discussions to offer valuable insights and interpretations of the obtained results. Overall, this chapter serves as a comprehensive repository, encompassing all the experimental data and analyses conducted throughout this study.
- **Chapter 5: Discussion** offers a comprehensive summary of the entire thesis work. We revisit and address all the research questions introduced in Chapter 1, providing detailed explanations and answers for each of them. We also discuss the limitations encountered during the research process, acknowledging any constraints or areas for improvement. Additionally, the future scope of the study is outlined, highlighting potential avenues for further exploration and development. Lastly, we provide a concise recap of the contributions made by our research, emphasizing the novel insights and advancements achieved through this thesis.
- **Chapter 6: Conclusion** summarize our thesis and provide a complete overview of our work. We highlight the experimental findings and explore their real-world applications. Furthermore, we discuss the exciting potential for future research in this field.

Chapter 2

Background and Related Work

Before diving into our main work, we have invested significant effort in conducting thorough background studies and gathering relevant research studies from the past. In this chapter, we have introduced the PmSys, a player monitoring system that provides daily updates on the wellness, illness, injury, and training load of soccer players. We also discussed the SoccerMon dataset and both subjective and objective metrics. Furthermore, we provided an overview of machine learning and the eight algorithms used in our experiments. We also discussed the sampling methods for machine learning. To address our imbalanced dataset, we utilized two different sampling techniques such as undersampling and oversampling. Additionally, we explored athlete health and monitoring. Later in this chapter, we presented information on the acute:chronic workload ratio and its application in identifying injury risk. We also discussed the use of artificial intelligence for injury prediction in soccer. Finally, we briefly reviewed the research conducted by other researchers in this field and highlighted their findings.

2.1 PmSys Framework

PmSys is a player monitoring system developed by the collaboration between researchers and students at Simula Research Laboratory, University of Tromsø, and ForzaSys [27] [51]. The system aims to assist athletes in their daily wellness reporting and provide trainers with valuable insights into player and team performance. The main component of PmSys is the mobile application. By using this application, athletes can easily report their daily wellness, injury, illness, game performance, and so on. The app is user-friendly and allows athletes to provide trainers with up-to-date information. The data collected from the app is then accessible through a web-based trainer portal. Trainers can access and analyze the reports submitted by athletes. This streamlined process saves time and effort, allowing trainers to have a holistic understanding of each player's fitness and well-being. The trainer portal provides a range of features to enhance the training and performance management processes. Trainers can use the data to curate training sessions that are tailored to individual players needs.

2.1.1 Mobile App

The PmSys mobile application is available on both Android and iOS platforms, making it easily accessible for athletes. This user-friendly app offers a wide range of options for athletes to report on different aspects of their well-being and performance. These features, including coronavirus check,

wellness, injury, game report, session RP, participation, and illness, provide a holistic approach to their athletic journey [14]. Figure 2.1. visually represents these features.

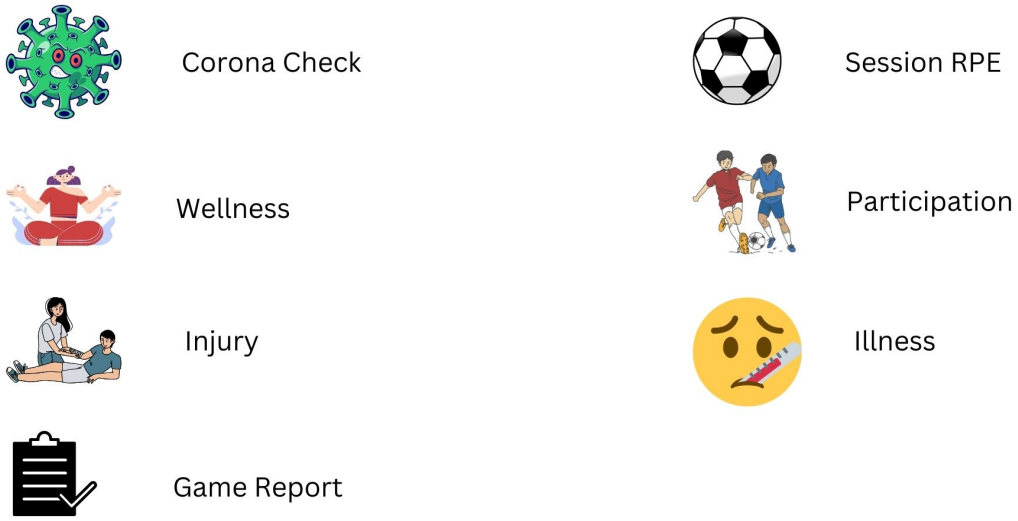

Figure 2.1: The figure indicates all the features of the PmSys mobile app's reporting features.

During COVID-19, athletes can easily report their completion of mandatory Corona checkups through a mobile app, providing crucial information to officials. The app also allows athletes to report their wellness, including sleep quality, mood, stress, soreness, and fatigue, enabling them to communicate their physical and mental well-being. In the injury section, athletes can report any health conditions to trainers and club officials, facilitating appropriate action. The game report function enables athletes to provide feedback after each match, assisting coaches in analyzing performance and planning future sessions. Additionally, athletes can subjectively assess their perceived exertion through the Session Rating of Perceived Exertion (sRPE) feature. The participation function allows athletes to report their involvement in activities, training, and exercise actions, while the illness section permits reporting of specific symptoms, aiding in the identification and management of potential health issues. Overall, this comprehensive mobile app facilitates efficient and effective communication between athletes and officials, ensuring the health, safety, and performance of athletes during these challenging times.

2.1.2 Web App

The web application is designed to streamline communication and data management for various stakeholders within a sports organization [14]. Coaches, physicians, trainers, and club authorities can easily access athlete reports submitted through the PmSys mobile app. These reports provide a comprehensive overview of athlete performance and well-being, allowing officials to make informed decisions. The web app offers visual representations of key metrics such as injuries, illnesses, and performance parameters, aiding coaches in formulating strategies and selecting the best team for upcoming matches. Club physicians can access injury and illness reports to provide tailored treatments and care plans for players. Additionally, club authorities can use the web app

to assess athlete performance for salary and contract negotiations. Overall, this user-friendly web app enhances communication and supports decision-making for all key stakeholders in the sports organization.

2.1.3 Soccer Dashboard

SoccerMon is a comprehensive dataset specifically focused on elite women's soccer teams in Norway. It consists of two main components: subjective metrics and objective metrics. The subjective metrics portion includes a staggering 54,485 reports, with a total of 529,963 entries manually recorded. On the other hand, the objective metrics section encompasses 10,075 measurement sessions, resulting in a massive 6,248,770,794 GPS positions recorded on the fields. Overall, the dataset is comprised of an astounding 106,229,103,498 data points, highlighting its extensive nature. To facilitate the analysis and visualization of this vast dataset, the researchers at Simula Metropolitan Center for Digital Engineering AS (SimulaMet) developed an innovative tool known as the Soccer Dashboard [48]. This web-based tool provides users with the ability to visualize and analyze the data from the SoccerMon dataset. With the Soccer Dashboard, researchers, coaches, and other stakeholders can gain valuable insights and make informed decisions based on the extensive data available.

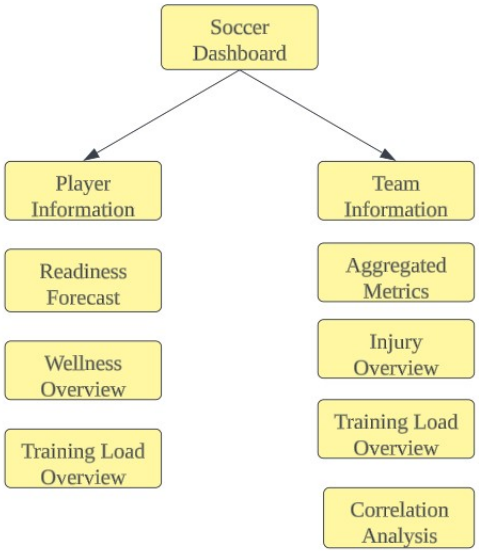

Figure 2.2: The features of Soccer Dashboard [48].

The Soccer Dashboard in Figure 2.2. offers a user-friendly and intuitive interface that allows users to easily access a wide range of player metrics, readiness and wellness forecasts, training load summaries, team metrics, injury insights, training load data, and correlation analyses for both Team A and Team B. This comprehensive overview provides valuable insights and information to help analyze and optimize player performance and team strategies.

The Figure 2.3. provides an easy and better visualization and analysis of individual women's soccer athlete readiness, wellness overview, and training load overview. The readiness forecast utilizes an auto-regressive moving average algorithm with exogenous regression, using historical data to anticipate a player's preparedness for training in the coming days. The Y-axis represents matrix

values ranging from 3 to 10, while the X-axis represents time.

The second analysis focuses on the wellness evaluation of athletes, considering parameters such as mood, sleep quality, stress levels, soreness, and fatigue. Coaches and trainers can access the wellness analysis report for individual athletes over different time frames, including the last week, fortnight, month, or even the last year. This comprehensive wellness analysis report offers a holistic view of an athlete's overall well-being. By examining these wellness parameters, coaches and trainers can make informed decisions tailored to the specific needs and conditions of each athlete.

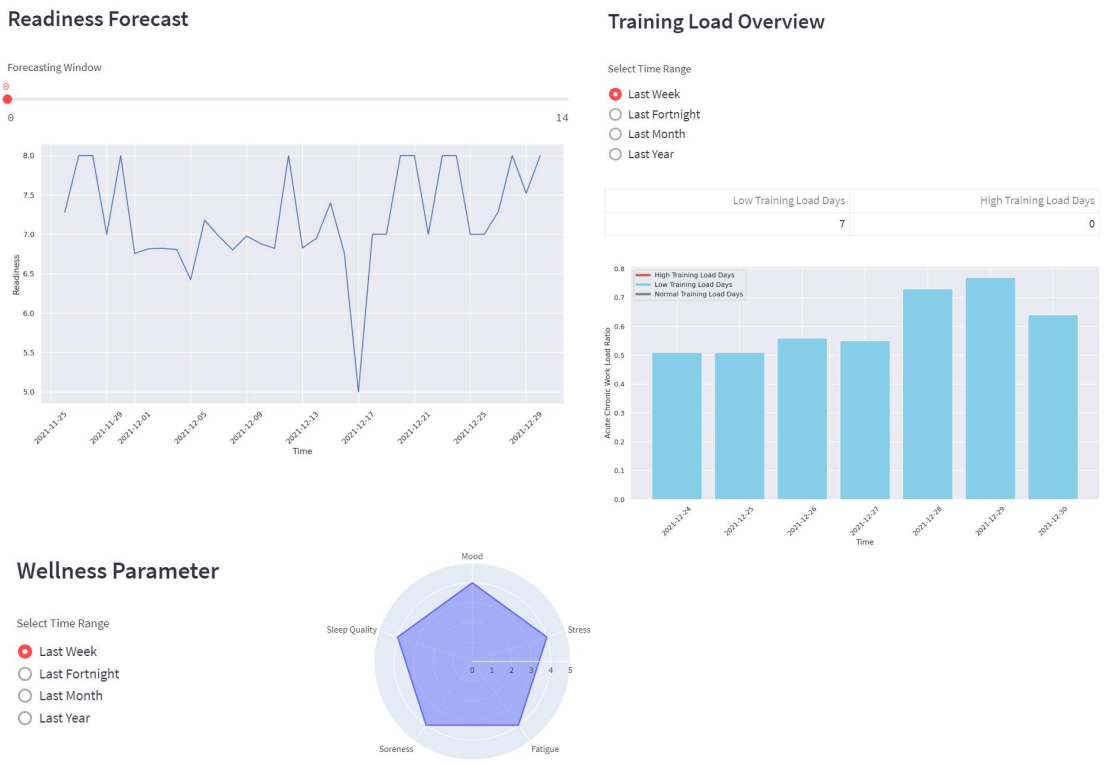

Figure 2.3: The figure highlights the information about the soccer players subjective metrics parameters [48].

The last analysis focuses on the athlete's training load overview, examining data from different time frames such as the last week, fortnight, month, or even the last year. The visualization graph represents the training load over time, with the X-axis indicating time and the Y-axis representing the acute chronic workload ratio (ACWR). A bar chart is used in the graph to display the variations in training load, making it easier to understand and interpret the data. The color scheme within the graph helps to interpret the intensity of the training load: red indicates a high training load, blue signifies a low training load, and black represents a normal training load. This visual representation provides a clear understanding of an athlete's training load status over time. Moreover, this analysis assists coaches in effectively managing and optimizing an athlete's training program. Furthermore, Coaches adjust training based on low and high training days to meet each athlete's needs.

	Mean ATL	Mean ACWR	Mean CTL28	Mean CTL42	Mean Strain	Mean Monotony	Mean Daily Load	Mean Session RPE
TeamA-d7299614-fa73-4f69-b5e9-f913e3154ff6	233.7300	0.6900	233.5900	233.5200	2,640.6200	1.0000	233.7300	203.1600
TeamA-b58af410-da77-479e-b93c-e03617b9f36d	259.6400	1.0300	256.4800	255.0800	2,339.2900	1.0700	260.5300	251.2500
TeamA-5cd7a61b-88b2-46d2-94f8-5a0d4f682d93	250.1800	0.5700	250.1800	250.1800	2,809.8500	0.7600	250.1800	233.5600
TeamA-74afe68c-f348-414c-9754-6d6f9df12587	324.6200	1.1600	321.0100	319.4000	3,699.5700	1.2700	326.0700	299.0700
TeamA-bcc03f81-2733-45d3-abf1-f7a709c63e68	332.8300	0.9000	331.6500	330.3200	3,823.9100	1.3500	332.8300	303.7500

Figure 2.4: The figure illustrates the visualization report for the teams [48].

The Soccer Dashboard provides an easy and comprehensive team-wide visualization and analytical report, as depicted in Figure 2.4. It includes aggregate metrics, injuries, training load data, and correlation analysis. The dashboard presents a visualization and analysis of training load and health indicators for each player within a chosen team and time frame. It also offers a comprehensive overview of the team's injury landscape, helping coaches identify strategies to mitigate injury risks. Additionally, the dashboard provides a tabulated report showcasing the training load of each player, aiding coaches in planning and scheduling future training sessions based on the collected data. Overall, the Soccer Dashboard serves as a valuable resource for coaches, providing insights and tools to optimize team performance and reduce the likelihood of injuries.

The Figure 2.5. presents the correlation matrix, which showcases the relationship between wellness parameters and training load metrics. This matrix is a helpful tool for analyzing the connection between different variables. It enables us to gain insights into how various matrices of wellness and training load parameters are interconnected. By examining this correlation matrix, we can better understand the relationship between these factors and their impact on performance and overall well-being.

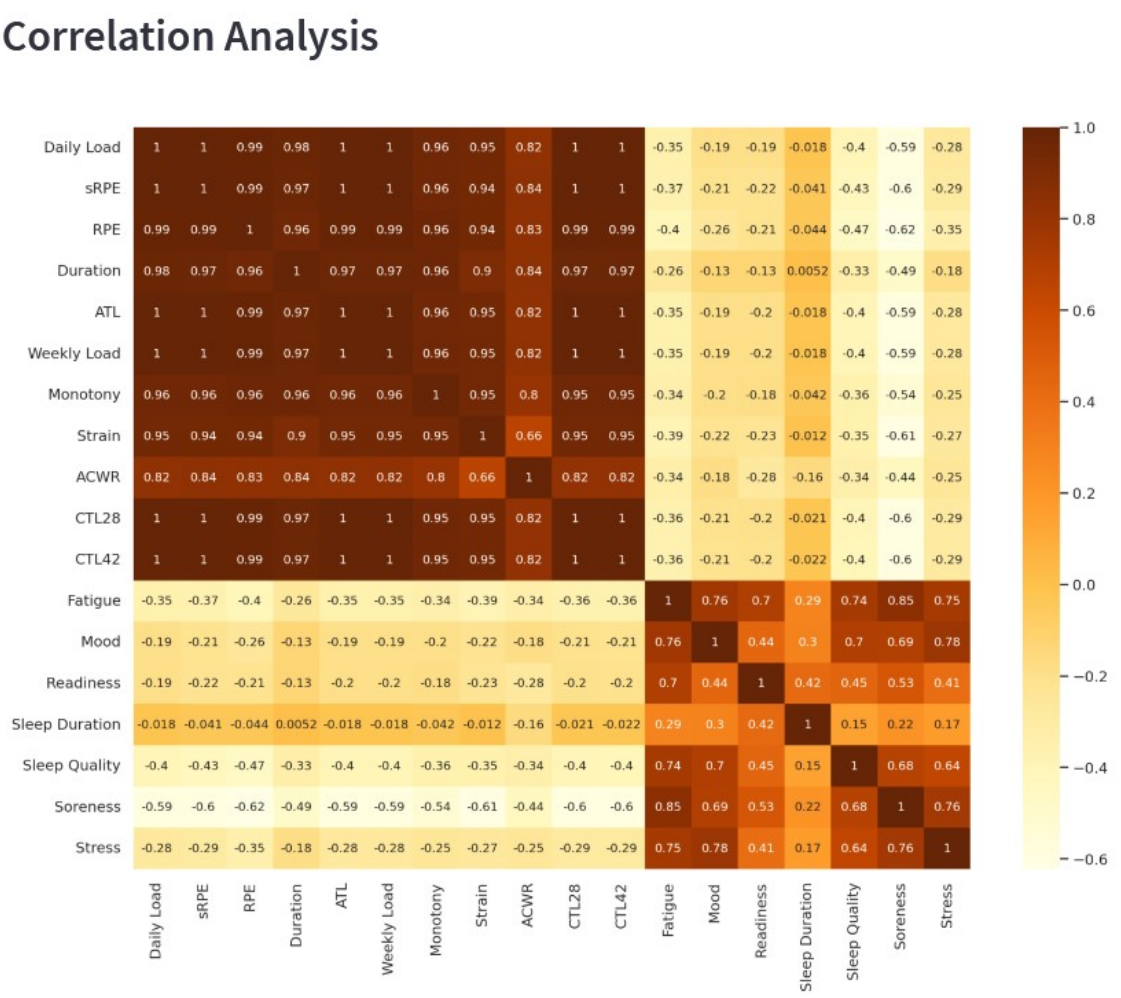

Figure 2.5: Correlation metrics from the Soccer Dashboard [48].

2.2 SoccerMon Dataset

The SoccerMon dataset [52], developed by researchers at Simula Research Laboratory, is a valuable collection of information on soccer players. This dataset was gathered over a span of two years from two elite women's soccer teams. It includes data on various aspects such as wellness, training load, illness, game performance, injury, and movements. The SoccerMon dataset is divided into subjective and objective metrics, with subjective metrics obtained through PmSys and objective metrics acquired from the STATSports APEX system.

The SoccerMon dataset is a valuable resource for researchers in the field of women's soccer. It is currently the largest dataset available, offering numerous possibilities for research and experimentation. While some studies have already been conducted using this dataset, such as the implementation of PmSys and predictions related to readiness and athlete performance, there is still much untapped potential. Researchers can use the SoccerMon dataset to develop strategies that specifically target injury prevention for women's soccer players. By doing so, they can greatly improve the well-being and performance of these athletes, ultimately making a significant impact on the sport as a whole.

2.2.1 Subjective Metrics

The SoccerMon dataset is split into two categories: subjective metrics and objective metrics. The subjective metrics section comprises data on the player's training load, wellness, game performance, illness, and injuries. This information is collected through the PmSys system, which was developed by Simula Research Laboratory, the University of Tromsø, and ForzaSys. PmSys is a performance monitoring system specifically designed for soccer players. It allows players to record their training load, wellness, game performance, illness, and injury details. The PmSys application is available on both iOS and Android platforms, making it convenient for players to report their data after each task.

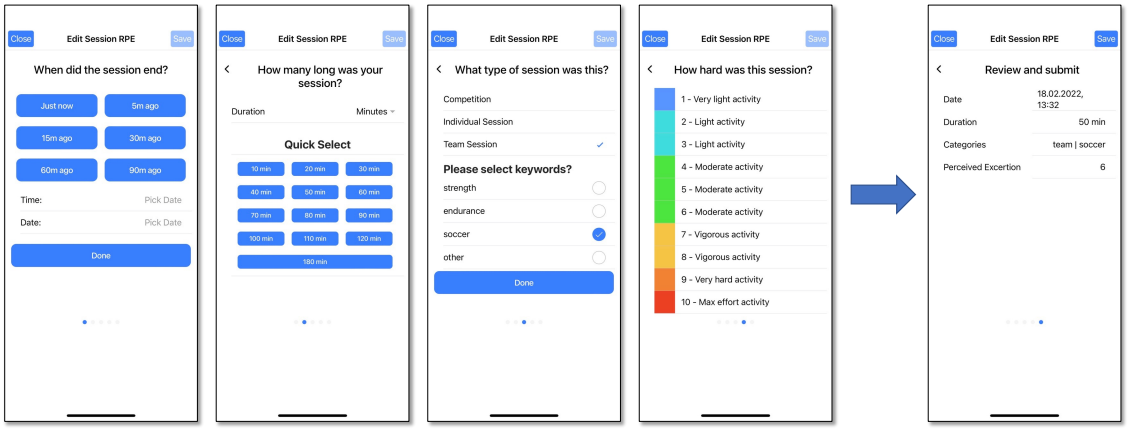

Figure 2.6: The illustration of sRPE reporting in PmSys [57].

The Figure 2.6. provides a visual representation of the sRPE (session rating of perceived exertion) in the PmSys application within the player monitoring system. Athletes engage in multiple training sessions or activities leading up to a match to adequately prepare themselves. Following each training session, athletes utilize the PmSys application to report their training load. During the

reporting process, athletes encounter various questionnaires, including details such as the duration of the training period, date and time of training, types of exercises performed, and the perceived difficulty level of the session on a scale of 1 to 10. Once all the questionnaires are completed, players submit their reports through the mobile app.

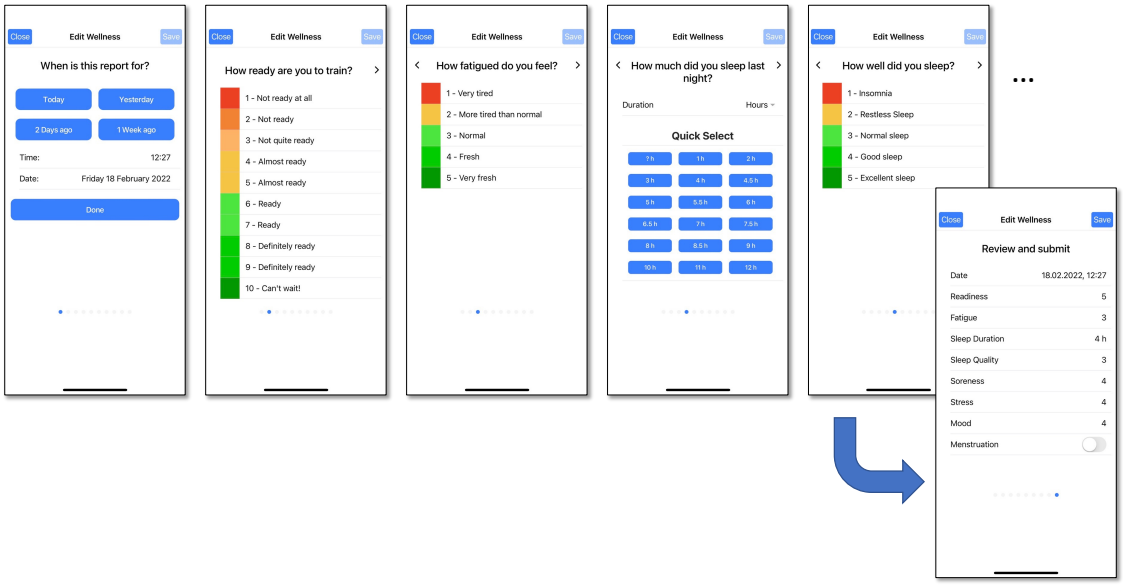

Figure 2.7: The illustration of wellness reporting in pmSys [57].

The Figure 2.7. presents the interface for reporting wellness within the player monitoring system. The parameters for reporting wellness include readiness, fatigue, sleep duration, sleep quality, soreness, stress, mood, and menstruation. The reporting process follows a similar method to that of reporting sRPE. The first question prompts athletes to select the date for which they are reporting their wellness. The second question revolves around the athlete's readiness for training, allowing them to indicate their readiness level on a scale from 1 to 10, with options such as ready for training, not ready, almost ready, and so on. The third question focuses on the athlete's general feelings. Following that, athletes are asked to report their sleep duration from the previous night. Lastly, athletes are prompted to provide feedback on the quality of their sleep from the previous night.

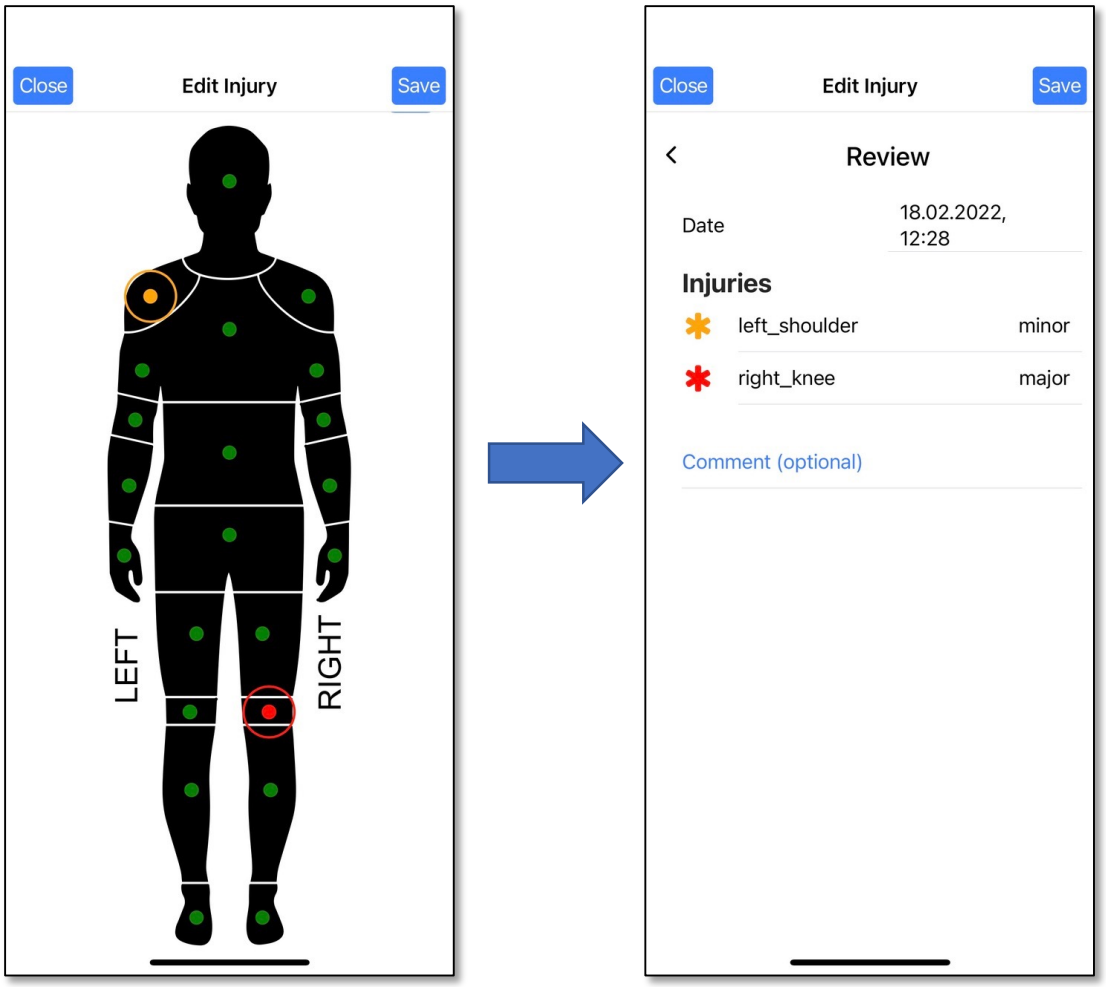

Figure 2.8: The illustration of injury reporting in PmSys [57].

Athletes often sustain injuries during their training sessions. To simplify the reporting process, the PmSys mobile app allows athletes to report their injuries conveniently. Figure 2.8. showcases the user-friendly interface of the app, specifically designed for injury reporting. By simply tapping on the affected area of their body, athletes can report the injury. The app uses a color-coded system, with yellow indicating a minor injury and red representing a major injury. This straightforward and intuitive process allows athletes to quickly and accurately report their injuries for proper management and treatment.

2.2.2 Objective Metrics

The SoccerMon dataset includes objective metrics collected using the STATSports APEX system. This system, developed by STATSports, provides valuable insights into an athlete's physical performance and health. It tracks various parameters such as player movement, heart rate, distance traveled, speed, acceleration, and more. The APEX system is approved by FIFA and is widely used by coaches and sports scientists. Athletes wear a GPS tracker vest, as shown in Figure 2.9, which is equipped with lightweight GPS and sensors to collect the necessary data. This data

helps in analyzing performance, making informed training decisions, and preventing injuries.

(a) input (b) output

Figure 2.9: STATSports GPS tracker [61].

The STATSports APEX system is not limited to specific leagues or countries. It is widely used by various sports teams around the world, including Premier League soccer clubs in the UK, National Rugby League (NRL) teams in Australia, and National Soccer League (NFL) teams in the United States. In the case of the women's elite soccer teams in Norway, researchers from Simula Research Laboratory, UIT The Arctic University of Norway, and Forzasys utilized the APEX system to collect objective metrics. After each training or game session, the data captured by the wearable units of the APEX system is retrieved and seamlessly uploaded to the club's dedicated laptop using the STATSports Sonra 2.1.4 software. The collected data is then systematically processed and stored as SoccerMon objective metrics for further analysis and insights.

2.3 Machine Learning

Machine learning, a subset of artificial intelligence, is frequently used to find patterns in data through the use of algorithms [45]. The algorithm constantly aims to increase its forecast accuracy by learning from the available data. There are several key components of machine learning, such as data, features, algorithms, training, testing, validation, hyperparameters, and so on. The concept of machine learning has been around for several decades. The idea was first introduced in the 1950s by a man named Arthur Samuel [53], who worked at IBM and was known as a pioneer in the field of artificial intelligence. Nowadays, many sectors are using machine learning techniques to increase productivity. The healthcare sector uses machine learning for drug discovery, disease prediction, and patient outcome prediction. Frauds are a big threat to the banking sector. Banks are using machine learning techniques to detect prospective fraud. Machine learning is also used for image classification, object detection, facial recognition, chatbots, and so on [29]. All of us have heard about the self-driving car. Self-driving cars use machine learning for perception and decision-making. In the following part, we will discuss some additional aspects of machine learning as well as algorithms that are relevant to this research study.

2.3.1 Supervised Learning

Supervised learning is a sub-field of machine learning and artificial intelligence. In supervised learning, algorithms are trained using labeled data to predict outcomes accurately [19]. Labeled data represents meaningful data attributes like name, type, or number. Supervised learning learns patterns and relationships between the input and output. Additionally, it applies algorithms to learn the relationship between features and targets from the given dataset. There are two types of supervised learning: classification and regression. Some of the supervised learning algorithms are Random Forest, Support Vector Machine, Linear Regression, Logistic Regression, k-nearest Neighbors, Naive Bayes, and Neural Networks. Supervised learning is mostly used in the finance, retail, manufacturing, agriculture, and marketing sectors. There is also huge potential in the fields of education, pharmaceuticals, and nutrition.

Supervised learning is widely used in the healthcare sector. It can help doctors analyze the X-ray image to detect tumors. The algorithms can predict the risk of various health-related outcomes, including heart disease, stroke, and readmission rates. It is particularly well-suited for injury prediction tasks because it allows for the development of predictive algorithms that can help identify potential risks and take preventive measures to reduce the occurrence of injuries.

2.3.2 Unsupervised Learning

Unsupervised learning is a type of machine learning used to analyze and cluster unlabeled datasets [43]. The main difference between supervised and unsupervised learning is that unsupervised learning can handle data without any label, whereas supervised learning is only applicable to labeled data. In unsupervised learning, users do not need to guide the algorithm; instead, the algorithm learns by itself and discovers patterns and information that were previously undetected. Some of the unsupervised learning algorithms are k-means clustering, hierarchical clustering, isolation forests, autoencoders, etc.

It is very popular for clustering, anomaly detection, dimensionality reduction, recommendation systems, natural language processing, genomics, video analysis, robotics, and autonomous systems. Moreover, unsupervised learning is used for climate data analysis to identify weather patterns and trends. Furthermore, it is used in the e-commerce sector to recommend products and content to customers to increase the volume of sales.

2.3.3 Logistic Regression

Logistic regression is a statistical method used for binary classification tasks, where the goal is to predict the probability that an observation belongs to one of two classes [25]. It works by estimating the probability that an observation belongs to a particular class based on its features. Unlike linear regression, logistic regression algorithms this relationship using the logistic function, which ensures that the predicted probabilities fall within the range of 0 to 1. Logistic regression uses a threshold (often 0.5) to compare these probabilities and classify observations into the most likely group.

Because logistic regression is easy to understand and effective at handling binary classification tasks, it is still widely used in various fields such as medicine, social sciences, marketing, and finance. It is also possible to use logistic regression for predicting customer churn [5], determining disease diagnosis, analyzing the impact of marketing campaigns, and more.

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_n X_n)}} \tag{2.1}$$

Formula 2.1 shows the formula for logistic regression, where, P(Y=1|X) is the probability of the dependent variable (Y) being 1 given the predictor variables (X), β0, β1, β2, ..., βn are the coefficients or weights associated with each predictor variable (X1, X2, Xn) and e is the base of the natural logarithm. In logistic regression, the coefficients (β0, β1, β2, ..., βn) are estimated through a process called maximum likelihood estimation. The goal is to find the best-fitting line that maximizes the likelihood of observing the given data.

2.3.4 Decision Tree

The decision tree is one of the popular machine learning algorithms that build a tree-like algorithm for making decisions or predictions by learning simple decision rules from the data [37]. It is widely used for both classification and regression tasks. The decision tree algorithm starts with a single node, known as the root node, which represents the entire dataset. It then splits the data based on the values of different features to create decision nodes. Each decision node represents a test on a specific feature, and the branches from the node represent the possible outcomes of that test. The process continues recursively until a stopping condition is met, such as reaching a maximum depth or having a minimum number of samples in each leaf node. To make predictions using a decision tree, new data is passed down the tree from the root node to the leaf nodes based on the test conditions at each decision node. The predicted outcome is then determined by the majority class or the average value of the samples in the corresponding leaf node. The equation for a decision tree in machine learning can be represented as follows:

$$y = f(x) = \begin{cases} y_1 & \text{if } x \leq t_1 \\ y_2 & \text{if } t_1 < x \leq t_2 \\ \vdots & \\ y_n & \text{if } x > t_n \end{cases} \tag{2.2}$$

There are several use cases for decision trees. It is commonly used for classifying data into different categories or classes, such as identifying different groups of customers based on their characteristics and behaviors, identifying fraudulent transactions or activities based on patterns and indicators, assessing patient symptoms and medical history to determine potential diseases or conditions [18], etc.

2.3.5 Random Forest

Random forest is one of the popular machine learning algorithms that combine several decision trees to make predictions [30]. It is a method of collaborative learning that improves the stability and accuracy of each decision tree separately. During the training process, random forests create a multitude of decision trees. Each single tree is trained on a random subset of the training data, and only a subset of the input features is considered at each split. Because of this process, random forest is good at handling overfitting as well as increasing the generalization ability of the algorithm [21]. To make a prediction using a Random Forest algorithm, each tree in the ensemble independently predicts the class label (for classification) or output value (for regression). The final prediction is determined by taking the majority vote (for classification) or the average (for

regression) of the predictions from all the trees. The equation for a random forest can be expressed as:

$$y = f(x) = \sum(w \cdot h(x)) \tag{2.3}$$

Here y presents the predictive outcome. f(x) is the function that predicts the outcome based on the input variables (x). Σ denotes the summation of all the trees in the random forest. w represents the weight or importance assigned to each tree's prediction. h(x) represents the prediction made by an individual decision tree based on the input variables. Random forest is useful in various applications such as classification, regression, and feature selection. It can handle larger datasets and high-dimensional data. Moreover, it can handle overfitting better than individual decision trees.

2.3.6 K-Nearest Neighbors (KNN)

KNN stands for K-Nearest Neighbors, which is one of the popular machine learning algorithms used for classification and regression tasks [22]. It works by finding the closest k neighbors to a given data point based on a distance metric, and then making predictions based on the labels or values of those neighbors. Consider a data point to be classified/predicted as P, and let D be the set of all data points in the training dataset. The KNN algorithm calculates the distance between P and each data point in D using a distance metric (e.g., Euclidean distance), and selects the k nearest neighbors of P. The class label y of P is determined by assigning the most prevalent class label among the k neighbors.

Euclidean Distance Formula: (2.4)

The Euclidean distance between two data points (x_1, y_1) and (x_2, y_2) in a two-dimensional space is given by:

$$\text{distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2.5}$$

This formula measures the straight-line distance between two points in a Cartesian coordinate system. It can be extended to higher-dimensional spaces by including the corresponding differences in coordinates.

2.3.7 LSTM

LSTM stands for Long Short-Term Memory, commonly used in various applications, such as natural language processing and time series analysis [41]. It is designed to effectively capture long-term dependencies in sequential data by incorporating memory cells and gating mechanisms [40]. The working process of LSTM:

Input Gate: The input gate determines which information from the current input should be stored in the memory cell. It uses the previous output and the current input as input, applies a sigmoid function to them, and produces a value ranging from 0 to 1 for each memory cell element. It is calculated as:

$$i(t) = sigmoid(Wi \cdot [h(t-1), x(t)] + bi) \tag{2.6}$$

Forget Gate: The forget gate decides which information from the previous memory cell should be discarded. It takes the current input and the previous output, applies a sigmoid function to them, and outputs a value between 0 and 1 for each element of the memory cell. It is calculated as:

$$f(t) = sigmoid(Wf \cdot [h(t-1), x(t)] + bf) \tag{2.7}$$

Update Memory: The memory cell is updated through the interaction of the input gate and forget gate. Which values should be updated and which should be forgotten are determined by the input gate and forget gate, respectively. The input gate value is multiplied by the current input and added to the forget gate value, which is multiplied by the value of the preceding memory cell, to complete the update. The update is calculated as:

$$g(t) = tanh(Wg \cdot [h(t-1), x(t)] + bg) \tag{2.8}$$

Output Gate: The output gate value is multiplied by the current memory cell state. This calculation determines which information should be passed as the output of the LSTM unit. It is calculated as:

$$o(t) = \sigma(W_o \cdot [h(t-1), x(t)] + b_o) \tag{2.9}$$

LSTM is widely used for text classification, sentiment analysis, speech recognition systems, weather forecasting, video captioning, handwriting recognition, etc.

2.3.8 Support Vector Machine (SVM)

The way it operates is by identifying the best hyperplane to divide the data points into several classes and categories. It aims to maximize the margin between the hyperplane and the nearest data points [1], which helps in achieving better generalization and classification accuracy. There are two different formulas for binary classification and regression. The SVM formula for binary classification can be represented as:

$$f(x) = \text{sign}(w \cdot x + b) \tag{2.10}$$

Here, $f(x)$ is the predicted class for a given input x, w is the weight vector, x is the input vector, $\cdot$ represents the dot product, and b is the bias term. The SVM formula for regression can be written as:

$$f(x) = w \cdot x + b \tag{2.11}$$

In the formula 2.9, $f(x)$ is the predicted value for a given input x, w is the weight vector, x is the input vector, $\cdot$ represents the dot product, and b is the bias term.

There are many benefits to using SVM, such as being effective in high-dimensional spaces, robust against noise and outliers, versatile in kernels, etc. Moreover, SVM is used in fields like bioinformatics, genomics, text and image classification, and anomaly detection.

2.3.9 XGBoost

XGBoost, also known as eXtreme Gradient Boosting, is a popular machine learning algorithm widely used for regression and classification tasks [47]. It works by gradually constructing a collection of weak decision tree algorithms. Every new algorithm that comes out is trained to fix the errors that previous algorithms made. This process is done iteratively, and the algorithms are combined using

a technique called gradient boosting. In each iteration, the algorithm calculates the gradient of the loss function concerning the predictions made by the ensemble so far. It then fits a new decision tree algorithm to the negative gradient of the loss function, effectively minimizing the loss. The predictions of this new algorithm are added to the ensemble, and the process is repeated until a stopping criterion is met. The formula of XGBoost can be divided into two parts: the objective function and the prediction function.

Objective Function: The objective function in XGBoost is a combination of a loss function and a regularization term [1]. It measures the quality of the algorithm's predictions and helps in optimizing the algorithm during training. The general form of the objective function is:

$$\text{Objective} = \text{Loss}(y, \hat{y}) + \lambda \cdot \text{Regularization} \tag{2.12}$$

Here, y represents the true labels, ŷ represents the predicted labels, and λ is the regularization parameter. The loss function quantifies the difference between the true labels and the predicted labels. Common loss functions include mean squared error (MSE) for regression tasks and log loss (also known as cross-entropy) for classification tasks.

Prediction Function: The prediction function in XGBoost combines the predictions of multiple weak decision tree algorithms. Each weak algorithm is a simple decision tree that predicts a specific output value based on a set of input features. The final prediction is obtained by summing the predictions of all the weak algorithms, weighted by a learning rate (η):

$$\hat{y} = \eta \cdot \sum (\text{prediction of each weak algorithm}) \tag{2.13}$$

The learning rate (η) in XGBoost controls the contribution or weight of each weak algorithm to the final prediction. It scales the predictions made by each weak algorithm before summing them up.

There are many use cases for XGBoost, such as predicting sales, customer churn, credit risk, search engine recommendation systems, sentiment analysis, text classification, etc.

2.3.10 Naive Bayes

Naive Bayes is a classification algorithm based on the Bayes theorem, which assumes that the presence of a particular feature in a class is unrelated to the presence of any other features [31]. It is referred to as "naive" since it strongly presumes that each feature is independent of the others. The Naive Bayes formula can be obtained by applying Bayes' theorem. The Naive Bayes algorithm, given a class variable C and a set of features X = x1, x2,..., xn, determines the probability of a given class given the features in the following way:

$$P(C|X) = \frac{P(C) \cdot P(X|C)}{P(X)} \tag{2.14}$$

Here, P(C|X) is the posterior probability of class C given the features X, P(C) is the prior probability of class C, P(X|C) is the likelihood of the features X given class C, P(X) is the probability of the features X.

Naive Bayes is used for recommendation systems to predict user preferences as well as item recommendations [11]. Banks and other financial institutions use it to identify fraudulent

transactions or activities based on various features. Moreover, Naive Bayes uses news filtering systems to categorize news articles and recommend relevant articles. Furthermore, it is widely used to analyze and classify opinions and sentiments expressed in text data, often on social media sites like Facebook and LinkedIn.

2.4 Sampling in Machine Learning

In the real world, we use different types of data to do analysis and prediction. Although the dataset may be extremely well organized, we often encounter datasets that require preprocessing before we can effectively use them. One common challenge is dealing with imbalanced datasets where one class is significantly more prevalent than the others. To overcome this challenge, sampling techniques are commonly used. These techniques aim to balance the class distribution and improve the accuracy of algorithms for minority classes. By manipulating the dataset, we can ensure equal representation of all classes and mitigate the bias towards the majority class. Sampling techniques play a vital role in addressing the issue of imbalanced datasets and enhancing the reliability of analysis and prediction tasks. In our thesis, we have used the undersampling and oversampling techniques to handle the class imbalance of our dataset.

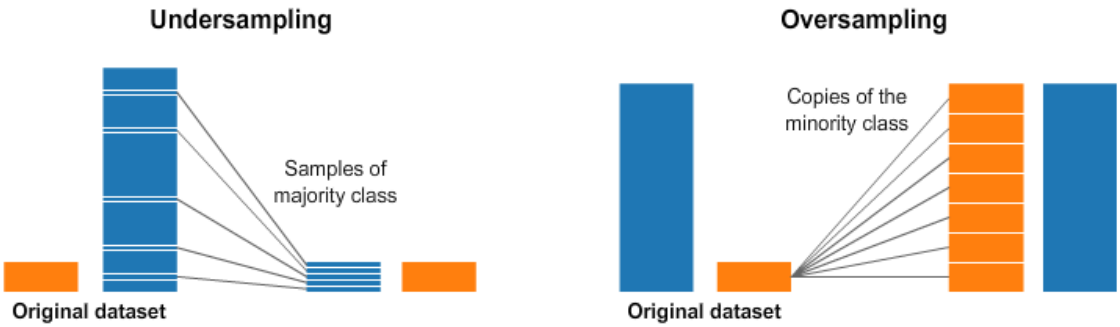

Figure 2.10: The figure represents the working methods of undersampling and oversampling [60].

2.4.1 Undersampling

Undersampling is another commonly used technique to handle imbalanced datasets [20]. Unlike oversampling, which increases the representation of the minority class, undersampling involves reducing the number of instances in the majority class to balance the class distribution [36]. The goal of undersampling is to create a more balanced dataset by randomly selecting a subset of instances from the majority class. By reducing the number of instances in the majority class, the class distribution becomes more balanced, allowing the algorithm to give equal importance to both the minority and majority classes during training. There are various undersampling techniques available such as random undersampling, cluster-based undersampling, and tomek links undersampling.

There are several benefits to using undersampling, such as the fact that the fact that it directly tackles the issue of class imbalance by reducing the number of instances in the majority class. This helps to create a more balanced dataset, allowing the algorithm to give equal importance to both the minority and majority classes during training. Moreover, by reducing the dataset size,

undersampling can lead to faster training and inference times. With fewer instances to process, the computational complexity of the algorithm decreases, making it more efficient.

2.4.2 Oversampling

One common method for dealing with imbalanced datasets in machine learning and data analysis is oversampling. A dataset is considered imbalanced if significant numbers of its instances belong to one class (majority class) and a smaller proportion to another class (minority class) [38]. The problem with imbalanced datasets is that they can lead to biased algorithms that perform poorly in predicting the minority class. This is because the algorithm is more likely to be biased towards the majority class due to its higher prevalence in the dataset. Oversampling helps overcome this issue by artificially increasing the number of instances in the minority class. There are a few different ways to perform oversampling. One common approach is to generate synthetic instances using techniques such as SMOTE (Synthetic Minority Over-sampling Technique). SMOTE works by creating synthetic instances by interpolating between neighboring instances of the minority class.

ADASYN also known as Adaptive Synthetic Sampling is another oversampling technique that is frequently used to address dataset imbalances [9]. It is an extension of the SMOTE algorithm that focuses on generating synthetic instances for the minority class in a way that is adaptive to the data distribution [24]. This algorithm addresses the limitations of SMOTE, which tends to generate synthetic instances indiscriminately and may not effectively capture the underlying data distribution. Considering the instance density distribution, ADASYN creates artificial instances in areas with lower densities. The advantage of ADASYN is that it helps to address the issue of overgeneralization that may occur with traditional oversampling techniques. By generating synthetic instances in a more targeted manner, ADASYN can better represent the complexities of the minority class and improve the algorithm's ability to accurately classify instances from the minority class.

2.5 Athlete Health and Performance Monitoring

Athlete health and performance monitoring indicates the collection and analysis of an athlete's physical condition and game performance. Athlete wellness monitoring has evolved to encompass a multidisciplinary approach. Sports organizations, trainers, and athletes are increasingly recognizing the value of complete wellness programs in optimizing performance and lowering the risk of injuries and illnesses. This section emphasizes the crucial importance of wellness in athlete health and performance.

2.5.1 Training

Effective training is the cornerstone of success in sports like soccer, cricket, and basketball. Soccer players, in particular, benefit from regular training sessions. Elite clubs such as FC Barcelona, Manchester City, Real Madrid, and Liverpool FC invest in top-notch training facilities and dedicated coaches to nurture their athletes. Manchester City has a total of 16 indoor and outdoor grounds like this where athletes can get training to prepare for the match. Proper training plays a pivotal role in translating practice into real match performance on the soccer field. Figure 2.11 shows one of the women's soccer players during the training session when they are preparing for the next match. A training session helps to collect data for a single athlete as well as for the team, which is very useful

for designing the training session for the coach. Moreover, it helps build muscular strength, power, and endurance through exercises and routines that target specific muscle groups. Furthermore, well-structured training helps to find out the potential athletes for the next game. In cricket, teams like India, Australia, and England have several small and large training grounds for their national as well as local teams. Each squad has its analyst who analyzes the athletes' training sessions to determine their strengths and weaknesses. They also assist in overcoming a player's weakness by evaluating his data more specifically. The trainer adjusts the athlete's training session based on the analytical results.

Figure 2.11: The figure illustrates the women's soccer player's training session [56].

2.5.2 Wellness

Athlete's wellness has a significant impact on the overall health and performance of individuals who participate in competitive sports and physical activities. It is an important component of athlete health and performance monitoring systems since it has a direct impact on an athlete's ability to train successfully and compete at their best. Physical health, mental health, emotional well-being, stress management, rest, and recovery are the primary components of athlete wellness. An athlete must be physically fit. She must maintain proper diet, hydration, physical fitness, and the avoidance and management of injuries and illnesses to accomplish this. Athlete wellness also addresses the mental stress of athletes. This includes stress management, motivation, focus, and psychological well-being. By managing the mental pressure and having a positive mindset for the game, you play an important role in doing well. Emotional health refers to an athlete's emotional stability as well as general happiness. Athlete wellbeing includes emotional support, resilience, and the ability to cope with disappointments. Moreover, it must monitor and guarantee that the

athlete gets adequate recuperation after each training session and competition. Without adequate rest, an athlete can sustain an injury that can take months to recover from. Furthermore, athletes are frequently under severe pressure, and appropriate stress management techniques are required to assist them cope with the demands of their sport. Mindfulness activities, meditation, and getting help from sports psychologists or counselors can all help with stress management.

2.5.3 Illness

In the realm of athlete health and performance monitoring, addressing the potential impact of illnesses on athletes is of paramount importance. Like other humans, athletes are not immune to illness. Athletes are susceptible to a wide range of illnesses including upper respiratory infections, gastrointestinal problems, and other common health problems. Moreover, they often face injuries in different areas of their body. It is very important to prevent the possibility of illness because an athlete is a big asset for a team as well as for a nation. Preventing illnesses in athletes is a proactive approach that involves various strategies. Proper hygiene, immunizations, and nutritional support are some preventive measures that can reduce the risk of illnesses.

It is always critical to discover any potential disease as soon as possible. Monitoring athletes for signs and symptoms, as well as using laboratory tests as needed, can help with early diagnosis. The early detection of a disease allows for timely action and treatment. In the modern era of sports science, data-driven methods to disease prevention and management are becoming increasingly important. The combination of smart technology and data analytics enables continuous monitoring of athletes' health and can provide early indications of potential problems, allowing for proactive solutions. Overall, appropriate management and recovery sessions can assist an athlete in recovering quickly. In addition, an effective communication agreement between athletes and health professionals is essential so that athletes may receive correct treatment without any complications.

2.5.4 Game Performance

The purpose of athlete health and monitoring is to optimize an athlete's ability to perform at their highest level during competitive events. Game performance is the output of an athlete's training, preparation, and mental and physical conditioning. Game performance is heavily reliant on an athlete's physical preparedness. Strength, endurance, speed, agility, and other physical attributes developed during training are put to the test during competition. Monitoring these characteristics enables coaches and athletes to fine-tune training programs and achieve optimal physical performance. Physical ability is only one aspect of gameplay; mental preparedness is equally important. Athletes have to cope with the psychological factors of competition, such as stress, nervousness, and concentration. Psychological assistance and therapies are critical in improving an athlete's mental state during competition.

A variety of performance indicators are used to evaluate game performance. Depending on the sport, these measurements can include things like points scored, shooting accuracy, running speed, distance covered, and much more. Performance metrics enable objective evaluation and aid in the identification of areas for development. Wearable equipment has transformed game performance monitoring in the current day. During competition, athletes may wear gadgets that track their movements, heart rate, and other physiological parameters. Real-time data can reveal player weariness, stress levels, and injury risk. Based on this information, coaches and support

personnel can modify their approach. Monitoring game performance is important not just for improving performance but also for preventing injuries. Overuse injuries, fatigue-related injuries, and acute injuries can all have an impact on an athlete's performance. Monitoring techniques can aid in the identification of potential injury hazards and enable interventions to lessen the likelihood of harm during games.

2.6 Injury Analysis and Prediction for Soccer

Injury is a major problem for all types of athletes. Soccer players frequently struggle with injury-related problems which have an impact on their everyday lives, performance in games, and careers. The club and national soccer teams suffer a lot from injuries [2]. An athlete can fall in injury during training or game time. There is a strong connection between training load, wellness, and game time [16]. It takes several days for an athlete to heal from an injury and return to sports. Moreover, club-injured athletes cost an extra budget for treatment, recovery sessions, and replacement. Many times, injuries force the team coach to change the game strategy, position, and tactics, which may affect the entire game. Furthermore, an injured athlete always has an emotional impact on her teammates. Nowadays, soccer clubs and research organizations collect data on soccer athletes during training sessions and game time to reduce the risk of injury. Organizations such as FIFA, Aspetar, and UEFA have several scientists who are working on injury prevention. To reduce the risk of injury, it is important to have a comprehensive injury prevention plan that includes rest and recovery, warm-up, proper training equipment, etc [3]. An injury prevention plan can help reduce the risk of injury significantly.

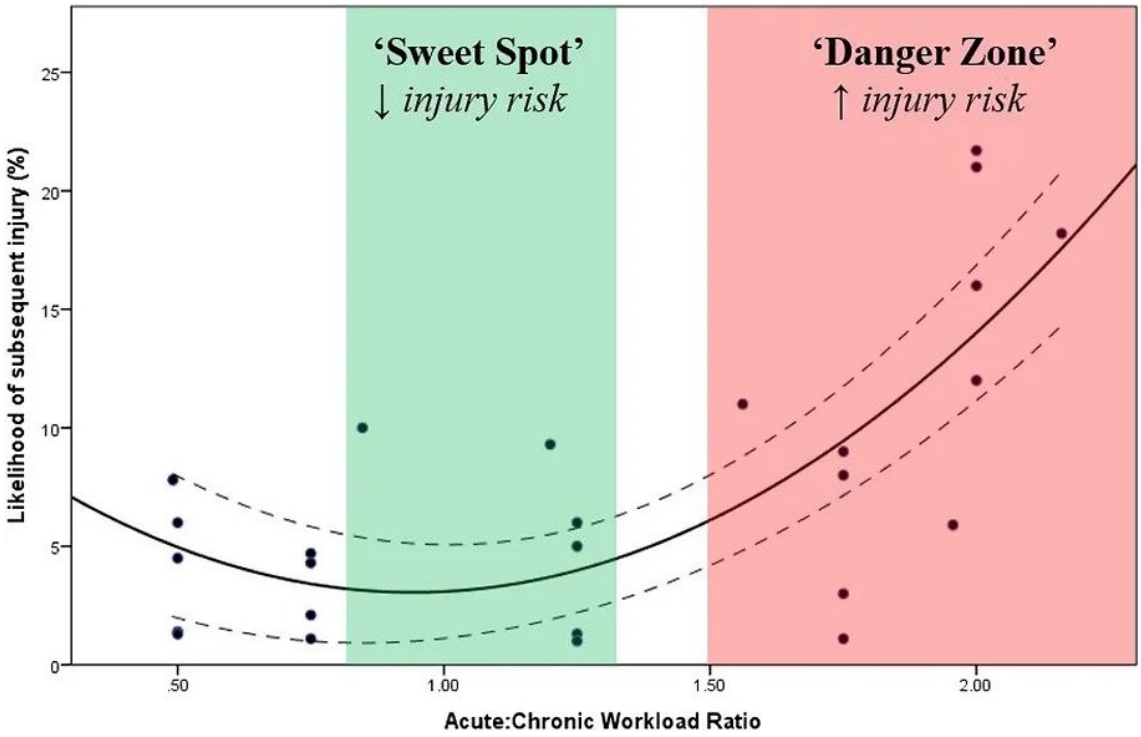

Figure 2.12: The Acute:Chronic Workload Ratio (ACWR) [17].

The Acute:Chronic Workload Ratio is a widely used metric in the field of sports science and medicine to assess injury risk among athletes, including soccer players [34]. It compares a short-term (acute) workload to a more extended (chronic) workload [35]. The ratio is calculated by dividing the acute workload by the chronic workload, providing insights into the balance between recent and long-term training stress. A lower or average ACWR indicates a low level of injury risk, while a high ACWR means there is a high chance of getting injured. The ratio of two distinct training periods is referred to as the ACWR. In ACWR, there are two parts the acute side and the chronic side. The acute side indicates the amount of training done by an athlete in the previous week, while the chronic side indicates the amount of training for the last four weeks. Therefore, the ratio is comparing the difference between the two. This helps to understand whether the training load is low or high. This helps to fix the training schedule of an athlete.

A paper written by the author, C. Want et al., discussed the effectiveness of ACWR [39]. They investigated the use of ACWR to comprehend more about how training load related to injury risks in athletes. Their primary goal was to highlight the relationship between levels of activity and the possibility of injury in the domains of sports science and medicine. While ACWR is commonly used to gauge a player's training load, some researchers have doubts about its effectiveness. According to the author FM Impellizzeri et al., there are many tools to calculate the training load, where ACWR is the most popular one. Their research found conceptual issues as well as fundamental limits associated with the use of this metric. The research is likely to examine differences and potential problems in understanding and implementing ACWR, calling into doubt its efficacy as a trustworthy tool for forecasting injury occurrences in sports science and medicine [35].

2.7 Application of AI for Injury Prediction in Soccer

Nowadays, artificial intelligence (AI) applications are widely used in different fields like industry, public health, education, the military, and even sports. In terms of sports, several teams and sports companies are using the AI application to increase their productivity and growth in the field. When we are talking about sports, soccer is a popular game all over the world, from Asia to America. In soccer, players are the most important asset of any club or team. But players often fall ill during training or match time. Here, AI has the potential to predict and prevent injuries to soccer players. This section provides an overview of the burgeoning field of AI in soccer injury prediction, highlighting its significance and potential implications for the safety and performance of players.

2.7.1 Transforming Injury Prediction

When it comes to sports injuries, traditional player examination techniques usually depend on the knowledge of medical professionals. Regretfully, these traditional methods frequently turn out to be expensive and time-consuming. Adding to the problem is the fact that there are times when a player gets hurt and the distance from the game site to the closest hospital becomes a major obstacle. Until the athlete can be sent to a hospital, the coaching staff has few options for getting up-to-date information regarding the player's status. AI offers a transformative paradigm by harnessing the power of data-driven analysis to provide a more accurate and comprehensive assessment of injury risk factors. Author Lars Hoel used soccer players GPS data to visualize and predict injury in this master's thesis. His work has proven that meaningful data can be used to forecast injuries and save a player's career [50]. Another author, Anna Linnea Jarmann, used survival analysis rather than the most common technique, ACWR, to identify the risk of injury factors [55].

2.7.2 Data-Driven Insights

AI technology helps to get information about a player's injury by analyzing data. The algorithms of AI process vast amounts of data, ranging from player biometrics and performance metrics to match and training conditions. This data-driven approach helps unlock the opportunity to drive into the reasons for injury, the condition of the players, and related information. Furthermore, the analysis report is useful to reduce the risk of injuries in the future. There are already several existing studies where researchers used data to predict not only the risk of injury but also the performance, readiness, and so on. Author Siarhei Kulakou and his team explored different time series algorithms to forecast the performance predictions of soccer players [42]. Another author, Mathias Menkerud Sagbakken, used the SoccerMon dataset to forecast the readiness to play of soccer players [48]. All this research suggests that machine learning techniques can be used to improve soccer and benefit players.

2.7.3 Injury Preventation Strategies

A soccer player often gets injured during training and, most often, in the game. During the training period, a player can suffer because of the long training period, hard exercise, and weather conditions. When it is game time, players can be tackled hard by the opposing team player, and there is a high possibility of injury. Here injury prevention strategies can play an important role in preventing injury. Coaches and medical personnel can reduce the chance of injuries by identifying soccer players who are more susceptible to injury and customizing training plans, rest periods, and preventive measures accordingly. These tactics can increase player lifespan and safety, which could improve soccer teams' overall performance.

2.7.4 Research Progress in AI for Soccer Injuries

Researchers are working on the injury factor of soccer players. Several clubs and national teams are interested in it, and the research scope and funding are increasing day by day. One of the well-known research organizations is the Female Football Research Center (FFRC), which works on women's soccer players health, sustainable development, and performance. Research organizations like FFRC are doing well because they are using the power of AI and other AI-related applications. There are many other research organizations, such as the FIFA Medical Assessment and Research Centre (F-MARC), the Union of European Football Associations (UEFA), and the International Olympic Committee (IOC), that work on injury prediction and prevention using different techniques of AI.

2.8 Overview of Related Work

Soccer has long been a popular sport around the world. Researchers are working on different aspects of soccer and players who play soccer for the national team, club, or league. Injury is one of the hot topics to do research on for all of the researchers. In this section, we will give a summary of some of the latest research on injury.

Injury Patterns and Prevention Strategies Among Elite Women's Soccer Players: Authors Astrid Junge and Jiri Dvorak have written a paper focused on data collection, types of injury, severity, affected body parts, and potential contributing factors [7]. According to them, there

are around 265 million soccer players, of which 26 million are women. Moreover, according to a FIFA report published in 2006, there are approximately 7 million women's soccer players in the United States. In Germany, the figure is 1.8 million, and for Mexico, it is 1.0 million. They also conducted a survey based on seven tournaments, where they found 387 injured players. Their research intends to identify the most common injuries, such as muscular strains, ligament sprains, and concussions. Moreover, based on their research, they likely propose modifications in training, rule changes, or injury management strategies to reduce the risk of injuries in games, tournaments, or training sessions. The primary objective of this study is to enhance comprehension of the patterns of injuries sustained by elite women's soccer players. This will facilitate the creation of better techniques for injury prevention and management that are specifically designed for this group of players.

Performance Enhancement in Adolescent Women's Soccer Players: Insights from a Ten-Week Injury Prevention Intervention: The paper titled 'A ten-week intervention in adolescent female football players' written by authors K. Steffen, H. Bakka, G. Myklebust, and R. Bahr discusses the effects of a ten-week injury prevention program on teenage women's soccer players [10]. Soccer has a greater injury ratio than any other physical game, including badminton, basketball, cricket, and so on, according to the authors' research. They conducted a ten-week experiment in Oslo, Norway, with thirty-four female participants. They mentioned in their paper that a program named "11" discussed injury prevention in soccer. The women's soccer players had to go through a variety of tests during the experiment days, including jumping, speed dribbling, shooting, a 40-meter single sprint, and more. This study focuses on the performance-related outcomes resulting from the implementation of the injury prevention program. Before and after the ten-week intervention, the data was gathered based on several performance characteristics, including strength, agility, and functional movement patterns. The analysis of the collected data is included in the paper, along with a discussion of the modifications or improvements to the players' performance metrics after the intervention. It most likely highlights how the injury prevention program affected performance factors, providing information on how to increase overall functional abilities, strength, and agility.

Physiological Demands and Tailored Training for Women's Soccer Players: A paper written by authors Naomi Datson and Andrew Hulton explores the physiological aspects of women's soccer players [13]. They studied the current knowledge on the topic and highlighted the updated information on the related research work. The authors placed high importance on the understanding of women's soccer players physiological demands. Moreover, they have discussed cardiovascular fitness, muscular strength, and endurance, highlighting the need for tailored training programs. The authors discuss gender-specific issues in sports medicine and support tailored strategies to maximize the performance of women's soccer players while lowering their risk of injury.

Financial Implications of Player Injuries in the English Premier League: There are several top leagues where many of the soccer players come from different parts of the world and play soccer for the club. Some of the great soccer leagues are La Liga, MLS, the English Premier League, UEFA, etc. The paper titled 'Estimation of Injury Costs: Financial Damage of English Premier League Teams' written by the author Eliakim E. et al. discussed the economic impact of injuries on English Premier League (EPL) teams [33]. The author's primary focus is figuring out the financial impact of player injuries that result in unsatisfactory performance. They utilize an extensive approach to evaluate the expenses connected to a team's poor performance as a result of player health

problems. The research investigation analyzes the financial impact of injuries on club performance using EPL data. The research offers important new information about the financial effects of player fitness on the performance and financial stability of Premier League teams by estimating injury-related financial damages. This research contributes to the understanding of the way player health, sports performance, and financial outcomes interact in professional soccer leagues.

Enhancing Soccer Athlete Performance Forecasting through Time Series Analysis: The master's thesis titled 'Soccer Athlete Performance Prediction Using Time Series Analysis' written by N. Ragab investigates the application of time series analysis in predicting the performance of soccer athletes [44]. This study aimed to forecast the future performances of soccer players using historical data. The author utilizes methodologies from time series analysis to develop predictive algorithms specific to soccer athletes. The research aims to contribute to the understanding of how temporal patterns in performance data can enhance predictive capabilities in the context of soccer. By exploring these predictive techniques, the thesis provides insights into potential advancements in performance management and training strategies for soccer athletes.

Another research paper titled 'Exploration of Different Time Series Algorithms for Soccer Athlete Performance Prediction' written by Siarhei Kulakou and his team used machine learning algorithms for soccer players performance. They have used the Norwegian women's soccer team data collected over two years for the prediction. Their research team focused on wellness parameters such as fatigue, sleep quality, and sleep duration to predict game performance. Among several machine learning algorithms, they used algorithms such as recurrent algorithms, algorithms of mixed recursive convolutional types, ensembles of deep CNN algorithms, and multivariate versions of the recurrent algorithms for prediction of performance [42].

Enhanced Monitoring of Illnesses and Injuries in Elite Soccer Players: The Oslo Sports Trauma Research Center Questionnaire: There are several methods for tracking soccer players' illnesses and injuries. Different clubs use different tools to monitor their soccer players. A paper written by the author B. Clarsen et al. introduced a new method to monitor the illness and injury of elite soccer players. The authors present the Oslo Sports Trauma Research Center Questionnaire, designed to systematically gather data on health issues among elite athletes. This questionnaire provides an expanded view of the well-being of players by offering an in-depth method for monitoring illnesses and injuries [12]. The creation and application of this instrument are discussed in the paper, with an emphasis on its potential for early wellness issue detection and prevention. The study emphasizes how crucial proactive monitoring is to optimizing player health and performance in competitive sports.

Insights into Health Challenges Among Youth Elite Athletes: The study conducted by C. Moseid et al., investigates the prevalence and severity of health problems in youth elite sports through a 6-month prospective cohort study involving 320 athletes [26]. The research highlights the comprehensive nature of the study, encompassing a diverse range of health problems that young elite athletes may face. The authors monitor and examine the participants' health problems during the period of the research by using a prospective strategy. The results throw light on the particular difficulties experienced by young athletes in elite sports and provide important insights into the prevalence and severity of health issues. To maximize athlete well-being and performance, the research highlights how important it is to understand and deal with health issues in this group. By providing an in-depth evaluation of the health environment in young elite sports, the research

makes a substantial contribution to the body of literature in sports science and facilitates the creation of focused treatments and preventive measures.

Enhancing Soccer Injury Studies: Consensus Framework for Standardized Definitions and Methodologies: Authors C. W. Fuller et al. discussed standardizing injury definitions and data collection methodologies in soccer studies in their paper titled 'Consensus Statement on Injury Definitions and Data Collection Procedures in Studies of Football Injuries' [4]. In their comprehensive consensus statement, the authors stressed several important aspects of soccer injuries, such as recurrent injuries, severity evaluation, and exposures during practice and competition. They also suggested a methodical classification system for injuries that considered the kind, location, diagnosis, and contributing variables. In addition to addressing the complex nature of soccer injuries, this all-encompassing strategy offers academics and practitioners a well-organized framework for data comparison and analysis. This comprehensive viewpoint emphasizes the value of industry-wide terminology and technique standardization, enabling a more nuanced knowledge of the various aspects of soccer-related injuries for enhanced management and preventative tactics.

Advancing Safety in Women's Soccer: Insights from a Meta-Analysis of Injury Prevention Programs: The safety of women's soccer players is very important and a matter of concern for every team and club. Authors Kay M. Crossley and Brooke E. Patterson wrote a paper focusing on enhancing safety in women's soccer. The study conducts a systematic review and meta-analysis, involving 11,773 women's soccer players, to evaluate the effectiveness of injury prevention programs [32]. The authors analyze existing literature to identify and assess interventions designed to reduce injuries in women's soccer. The meta-analysis provides quantitative insights into the impact of various preventive measures. The findings provide important information for the formulation of tactics meant to reduce the risk of injury for women's soccer players. Overall, the paper addresses a critical aspect of player welfare and safety in women's soccer through a comprehensive analysis of injury prevention programs.

Understanding Injury Triggers in Soccer Players: A Comprehensive Review of Gender-Specific Perspectives: Francesco Aiello and Franco M. Impellizzeri have written a systematic review paper that investigates injury-inciting activities in male and women's soccer players. The study includes a thorough examination of the body of research on the causes of sports-related injuries in male and female athletes [46]. The authors look at a variety of activities that can cause injuries, illuminating particular motions or situations that increase the danger of injuries in soccer. To offer an in-depth analysis of the factors impacting injury rates among male and female players, the review considers a variety of injury categories. It highlights how crucial it is to take gender into account while managing and preventing injuries. They consider 64 studies and point out 56,740 injuries. From their research, they found that high-intensity running and kicking are the most common reasons for the injury.

In conclusion, the vast literature analysis presented here highlights the collaborative effort of academics to understand the complexities of injuries among women's soccer players. The studies, which range from injury types and severity to preventative measures, greatly contribute to our understanding of athlete well-being. The authors not only identified common injuries but also made practical recommendations to reduce risks based on actual research. My thesis, "Predicting

the Injury Risk of Women's Soccer Players Using Machine Learning," strives to push the frontiers of injury prediction and prevention as this body of research establishes a solid basis. The goal of harnessing the power of modern computational methodologies is not only to build on previous research findings but also to introduce a forward-thinking approach that embraces the promise of machine learning in the field of women's soccer.

2.9 Chapter Summary

In this chapter, we provided a comprehensive overview of the player monitoring system (PmSys), which is the primary data source for SoccerMon. PmSys has both mobile applications as well as web-based versions for the users. Moreover. We also discussed the SoccerMon dataset we used to predict injuries. The dataset has both subjective as well as objective metrics. The subjective metrics include Norwegian elite two soccer team players wellness, training load, illness, game performance, and injury data, whereas GPS data belongs to objective metrics.

Moreover, this chapter presents information about the fundamental knowledge of machine learning that is required to build our machine learning framework to forecast injury. In addition, we have discussed Supervised Learning, Unsupervised Learning, Logistic Regression, Decision Trees, Random Forests, K-Nearest Neighbors, LSTM, Support Vector Machines, XGBoost, and Naive Bayes. In our studies, we have used these eight machine learning algorithms in the later chapters to predict the risk of injury for women soccer players. Furthermore, we have explored the athlete health and performance monitoring techniques that clubs are using nowadays for players and their benefits. Several clubs are tracking and analyzing player training, wellness, and game performance data to keep the players fit for the next match as well as prevent injury.

Sampling techniques are often used to handle imbalanced data. The main goal of this thesis is to predict injuries, although we have very few injury records. To handle the class imbalance and maximize the performance of the performance of the machine learning algorithms, we have utilized the undersampling and oversampling techniques. In the later part of this chapter, we have discussed one of the common metrics named ACWR used to assess the risk of injury in the sports science and medical sectors.

AI is widely used in different fields, and AI technology is becoming popular in sports, especially soccer. In this chapter, we have discussed the use of AI technology to prevent injury in soccer as well as other possibilities. Lastly, we have presented some previous work done by other researchers relevant to our thesis and highlighted their findings. In the next chapter, we have proposed a pipeline for our machine learning framework to predict the risk of injury.

In the next chapter, we discussed the methodology and implementation part of predicting Norwegian women's soccer players injuries. In the meantime, we have discussed data preprocessing, exploratory data analysis, feature engineering, model training, injury prediction, hyperparameter tuning, and so on.

Chapter 3

Methodology and Implementation

In the previous chapter, we presented the necessary background information and related research that is pertinent to our thesis. We started by introducing the PmSys framework and then delved into several important topics. These topics encompassed an analysis of the SoccerMon dataset, both subjective and objective metrics, a thorough exploration of various machine learning algorithms, different sampling techniques in machine learning, a detailed examination of athlete health and monitoring metrics, and an investigation into the application of AI for injury prediction in soccer. Our goal was to provide a thorough and comprehensive understanding of the fundamental components and concepts that form the basis of our thesis.

In this chapter, we presented a simplified and reader-friendly explanation of our methodology and implementation. We focused on the tools and techniques used for tasks such as data processing, exploratory analysis, feature engineering, and injury prediction. We broke down the steps involved in preprocessing the dataset, handling null values, analyzing the data, extracting features, implementing machine learning algorithms, model training, performance evaluation, and hyperparameter tuning. We also provided clear and straightforward explanations of the Python libraries and code used, making it easier for readers to understand and apply our approach to predicting soccer player injuries.

3.1 Proposed Pipeline

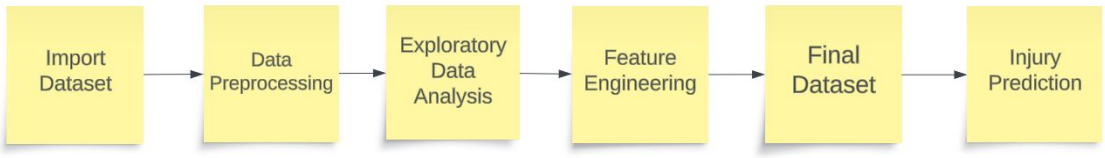

Figure 3.1: The flowchart represents the proposed pipeline for this thesis.

As shown in Figure 3.1. we have presented the proposed pipeline and will now provide a detailed explanation below.

Import Dataset: We have used the Pandas library to import the data mentioned in Section 3.2. The process of importing data involves bringing external data into a software application or

database. We used Pandas' read_csv() function, which simplified the import process for CSV files. This function allowed us to handle missing values, select specific columns, and perform other operations as needed. The imported data was then stored in a Pandas DataFrame, providing us with efficient capabilities for data manipulation and analysis.

Data Preprocessing: Data preprocessing played a crucial role in our study as it involved tasks such as cleaning, transforming, and organizing raw data to make it suitable for machine learning algorithms. This process was akin to preparing ingredients before cooking, as it set the foundation for achieving better results when building algorithms. In Section 3.3, we discussed the specific steps we followed to preprocess the data, which included addressing missing and duplicate values. By effectively handling these issues, we ensured that the data was of high quality and ready for analysis.

Exploratory Data Analysis (EDA): Exploratory Data Analysis (EDA) played a crucial role in our study as it involved analyzing, cleansing, manipulating, and applying algorithms to the data in order to extract meaningful information, identify patterns, uncover trends, and gain valuable insights. We have used various Python libraries such as Matplotlib, Seaborn, and Pandas for EDA, as mentioned in Section 3.4. These libraries provided us with a range of functionalities for visualizing and analyzing the data, enabling us to gain a deeper understanding of the dataset.

Feature Engineering: In the realm of data analysis, we have used the feature engineering technique to extract valuable information from existing features and create new ones. This technique allowed us to enhance our dataset by generating new features that contributed to improved algorithm performance. Through deriving meaningful insights from the data and creating relevant features, our goal was to optimize the accuracy and effectiveness of our machine learning algorithms. In Section 3.5, we provided a brief overview of feature extraction, feature scaling, aggregated metrics, sorting data, and other techniques employed in our study.

Final Dataset: To conduct our experiments and predict injuries, we created a total of three datasets. In Section 3.6, we outlined the process we followed to create these datasets. We provided a detailed explanation of how we collected and organized the data, as well as the criteria we used to filter and select the relevant information. By following this process, we ensured that our datasets were representative and suitable for training and evaluating our machine learning models.

Injury Prediction: One of the primary objectives of our thesis was to utilize machine learning techniques to predict injuries in women's soccer players. To accomplish this, we have used eight different machine learning algorithms to predict injuries. In Section 3.7, we have briefly discussed how we implemented our machine learning framework, training process, performance evaluation, and hyperparameter tuning. In this section, we have also discussed sliding windows and the class imbalance technique.

3.2 Import Dataset

In this section, we prioritized the selection of tools and techniques that would enable us to seamlessly integrate datasets into our pipeline. We placed a strong emphasis on robust data handling and accessibility, particularly when dealing with diverse sources such as files or databases. To achieve this, we carefully chose our tool set. It was crucial to ensure that all necessary

dependencies were installed prior to loading the dataset, as this ensured smooth and efficient data processing. By making deliberate choices and taking necessary precautions, we aimed to establish a solid foundation for handling and integrating datasets effectively.

For dataset management, we relied on the Pandas library, which is widely recognized for its flexibility and efficiency in data manipulation. To access our SoccerMon dataset, which was stored in Google Drive, we conveniently mounted Google Drive and utilized Google Cloud libraries for seamless data retrieval. In particular, we have used the pandas.read_csv() function to read CSV files and efficiently load data directly from Google Drive. This approach allowed us to streamline the process of accessing and managing our dataset, ensuring smooth data integration into our pipeline.

3.3 Dataset Preprocessing

Data preprocessing plays a crucial role in machine learning projects as it involves transforming raw data into a clean and understandable format before it can be utilized by machine learning algorithms. After importing the dataset, we performed a series of steps to cleanse and enhance the data before proceeding to feature extraction. These preprocessing steps are essential to ensure that the data is in a suitable state for analysis and modeling, thereby improving the accuracy and effectiveness of the subsequent machine learning algorithms. By addressing issues such as missing values, outliers, and data inconsistencies, we can ensure that the data is of high quality and ready for further analysis.

The SoccerMon dataset includes both subjective and objective metrics. We initially focused on the subjective metrics, which consisted of features like stress levels, daily workload, game performance, and soreness, among others. In this section of our thesis, we highlight the importance of data preprocessing in handling these subjective metrics. By applying techniques such as handling missing values, normalizing data, and addressing outliers, we ensure that the subjective metrics are properly managed and contribute effectively to our analysis and modeling.

Initially, each feature from the subjective metrics set was stored in separate CSV files. Our primary goal was to merge these files into a single unified CSV dataset. By employing effective data preprocessing techniques, we were able to successfully consolidate all the features from the individual CSV files into a cohesive and comprehensive dataset. This consolidation process ensured that all the necessary information from the subjective metrics was combined and ready for further analysis and modeling.

Furthermore, we integrated objective metrics from the work of Mathias Menkerud Sagbakken, which included GPS data [59]. This incorporation of objective metrics served to enhance our project by providing additional valuable information for analysis and modeling. By integrating both subjective and objective metrics, we obtained a more comprehensive dataset that allowed for a more thorough examination of factors influencing injury prediction in women's soccer players.

Additionally, a substantial portion of the dataset contained NaN (Not a Number) values. Addressing these NaN values is crucial to ensure the precision, reliability, and effectiveness of data analysis and machine learning endeavors. Employing appropriate strategies, such as imputation or deletion, is essential for resolving potential issues caused by NaN values in datasets. By handling NaN values

effectively, we can ensure that our data is complete and suitable for further analysis and modeling tasks.

In the SoccerMon Dataset, each feature was stored in a separate CSV file. To streamline further analysis, we combined these files into a single dataset. We used the pandas merge command for most of these files. However, due to the unique structure of the 'injury.csv' and 'illness.csv' files, we utilized a specific function to integrate these files into the main dataset. The details of this function are provided below:

```
1
2  def addNewDf(filename, datecolumn, newcolumn):
3      df2 = pd.read_csv(os.path.join(directory, filename))
4      df2.rename(columns={'player_name': 'player_names'}, inplace=True)
5      columns_to_keep = ['player_names', datecolumn]
6      df2 = df2[columns_to_keep]
7
8      merged_df = final_df.merge(df2, left_on=['Date', 'player_names'], right_on
       =[datecolumn, 'player_names'], how='left')
9      merged_df['timestamp'].fillna(0, inplace=True)
10     merged_df[newcolumn] = (merged_df['timestamp'] != 0).astype(int)
11     merged_df.drop('timestamp', axis=1, inplace=True)
12
13     return merged_df
```

In the SoccerMon dataset, each feature was initially stored in separate CSV files. To facilitate further analysis, we merged these files into a single dataset. We primarily used the pandas merge command for this task. However, due to the unique structure of the 'injury.csv' and 'illness.csv' files, we had to use a specific function to integrate these files into the main dataset. This function allowed us to merge the 'injury.csv' and 'illness.csv' files with the main dataset based on the 'Date' and 'player_names' columns. It ensured the correct updating of the new column, filled in missing values with 0, and converted matches into binary indicators. This approach was necessary because the 'injury.csv' and 'illness.csv' files had a unique dataset structure that required more specific handling compared to other files. By using this function, we were able to seamlessly integrate all relevant information into our final dataset. This ensured the integrity and completeness of our data for further analysis and model training.

3.3.1 Import Datasets

Initially, we imported all datasets using the pandas to_csv command. To merge them into a single dataset, we used a for loop to iterate through and read all the CSV files. The following code provides an insight into this process:

```
1
2  directory = '/content/drive/MyDrive/Soccer Player Injury Prediction/Datasets/
       subjective metrics/Data'
3  file_names = ['acwr.csv', 'atl.csv', 'ctl28.csv', 'ctl42.csv', 'daily_load.csv'
       , 'fatigue.csv', 'monotony.csv', 'mood.csv', 'readiness.csv','
       sleep_duration.csv', 'sleep_quality.csv', 'soreness.csv', 'strain.csv', '
       stress.csv',  'weekly_load.csv']
4  concatenated_columns_df = {file_name.split('.')[0]: pd.read_csv(os.path.join(
       directory, file_name)).iloc[:, 1:].values.flatten() for file_name in
       file_names}
5  concatenated_columns_df
```

The provided script starts by defining the directory containing the CSV files and listing the file names. It then initializes an empty DataFrame to store the merged data from all the CSV files. The for loop reads each CSV file, drops the first column that is not needed, and concatenates the remaining columns into a single column, which is then named after the file. This method allows us to efficiently combine multiple datasets into one, creating a comprehensive dataset for further analysis and model training. By unifying the data in this way, we can ensure a more streamlined and effective data analysis process.

3.3.2 Subjective Metrics

The subjective metrics contain raw information about players' daily activities. Once these were merged into a single dataset, our first challenge was to deal with a significant amount of null values. Handling these null values is a crucial step in data preprocessing as it ensures the integrity and reliability of the dataset for further analysis and machine learning algorithms.

Null Value Handling

To address the null values, we started by dropping 731 rows where the Date column was null, since date information is crucial for further procedures. This was accomplished using pandas' 'dropna' function. Next, we removed rows where all feature columns had null values and the injury column was 0. The following code illustrates this:

```
1
2 cols = ['fatigue', 'mood', 'readiness', 'sleep_duration', 'sleep_quality', '
    soreness', 'stress', 'team_performance', 'offensive_performance', '
    defensive_performance']
3 df = df.drop(df[df[cols].isnull().all(axis=1) & (df['injury'] == 0)].index)
4 print(df[df[cols].isnull().all(axis=1) & (df['injury'] == 0)].sum())
```

This procedure significantly reduced the number of null values in each column. Following this, we used the Iterative Imputer with a Random Forest Regressor from 'sklearn' to handle the remaining null values. This method iteratively estimates missing values by modeling each feature as a function of other features, which helps preserve the statistical properties of the dataset:

```
1
2 cols = ['fatigue', 'mood', 'readiness', 'sleep_duration', 'sleep_quality', '
    soreness', 'stress', 'team_performance', 'offensive_performance', '
    defensive_performance']
3 df[cols] = IterativeImputer(estimator=RandomForestRegressor(), random_state=0).
    fit_transform(df[cols])
```

By implementing this imputation process, we ensured that the dataset was free from missing values. This is a crucial step in data preprocessing, as it increases the robustness and accuracy of the subsequent model training phase. It allows for a more comprehensive analysis and enables the machine learning algorithms to perform optimally. With no missing values, the dataset became fully ready for the next steps in our machine learning pipeline, including exploratory data analysis, feature engineering, and model development.

Duplicate Rows

Once the null values were addressed, our next concern was handling duplicate rows. We identified and dropped duplicate rows using the duplicated() function. By removing duplicates, we ensured

that each observation was unique, which was vital for the integrity of our statistical analyses and machine learning algorithms. This step helped to mitigate any potential skewing of model training and improved the accuracy of injury predictions.

```
df = df.drop_duplicates()
```

By applying these steps, we ensured the integrity and quality of our dataset. We meticulously preprocessed the data to prepare it for subsequent analysis and model training, ultimately contributing to the accuracy and reliability of our injury prediction algorithms. With a clean and well-structured dataset, we could proceed confidently to the feature extraction and machine learning phases of our project.

3.3.3 Complete Dataset

Our next crucial step was to integrate the subjective metrics with the GPS data from the objective metrics. The objective metrics provided valuable insights into players' speed, HIR (High-Intensity Running), and other calculated metrics. We tackled this integration using the following code:

```

cols = ['Date', 'player_names', 'injury', 'acwr', 'atl', 'ctl28', 'ctl42', '
    monotony', 'strain', 'team_performance', 'offensive_performance', '
    defensive_performance', 'illness']
merged_df = pd.merge(df1[cols], df0[cols], left_on=['Date', 'player_names'],
    right_on=['date', 'player_name_x'], how='inner')

if not merged_df.empty:
    df0.update(merged_df[merged_df['Date'].eq(merged_df['date']) & merged_df['
    player_names'].eq(merged_df['player_name_x'])][cols[2:]].applymap(str.
    capitalize))

df0['Injury'] = merged_df['Injury']
df0.dropna(subset=['Injury'], inplace=True)
df0.reset_index(drop=True, inplace=True)
```

To integrate the subjective metrics with the GPS data from the objective metrics, we employed a script that extracted relevant columns from both datasets and merged them based on common identifiers such as date and player name. This process ensured consistency between corresponding columns from each dataset.

After merging the datasets, we iterated through the merged dataset to update the 'Injury' column in the objective metrics with the merged values. This step allowed us to incorporate the injury information from the subjective metrics into our objective metrics.

To ensure data quality, we also removed any rows with missing injury information. By performing these steps, we were able to create a unified dataset that combined both subjective and objective metrics, enabling us to conduct more comprehensive analyses and make more accurate injury predictions.

By executing this merging process, we successfully consolidated the subjective and objective metrics into a single dataset. This consolidation allowed us to perform a comprehensive analysis and develop models with a holistic view of player performance and injury-related factors. By combining the subjective metrics with the extracted features from GPS data, we created a complete

dataset that enhances the accuracy and effectiveness of our injury prediction algorithms. This comprehensive dataset serves as a valuable resource for further analysis and model development.

3.4 Exploratory Data Analysis

To gain a better understanding of our data, we used different visualizations. These visualizations helped us analyze the data from various angles and discover patterns, trends, and relationships. By presenting the information in a visual format, we were able to easily comprehend complex data and make more informed decisions. Overall, visualizations played a crucial role in enhancing our understanding of the data.

3.4.1 Correlation Analysis

Using Panda's corr() function and visualizing the results with a heatmap, we explored the data correlations. Our observations revealed strong positive correlations between certain variables, indicating that they tend to increase or decrease together. We also discovered negative correlations, where one variable increases while the other decreases. Additionally, we identified independent variables that showed no significant correlation with others. The heatmap provided a visual representation of these correlations, allowing us to easily identify patterns and relationships within the data. Overall, this analysis enhanced our understanding of how different variables are interconnected and influenced each other. Here is a summary of our observations:

Highly Correlated Features with Injury: The features that show a high correlation with injury, indicating a stronger relationship, include illness (0.065), weekly_load (0.059), ctl28 (-0.110), atl (-0.110), ctl42 (-0.105), monotony (-0.103), strain (-0.071), and acwr (-0.069). These features, along with injury itself (1.0), have a correlation coefficient above 0.05 or below -0.05, suggesting they may be significant predictors or indicators of injury occurrences.

Moderate Correlated Features with Injury: The moderately correlated features with injury have correlation coefficients ranging from 0.02 to 0.05. These include daily_load (0.031), fatigue (-0.029), average_running_speed (-0.030), soreness (-0.044), and top_speed (-0.044). While their relationship with injury is noticeable, it is weaker than that of highly correlated features, suggesting they may still contribute to injury prediction but with less impact.

Weakly Correlated Features with Injury: Features with weak correlations with injury, having correlation coefficients less than or equal to 0.02, include stress (0.002), defensive_performance (-0.002), Total_distance (-0.004), readiness (-0.006), injury_ts (-0.006), team_performance (-0.007), mood (-0.008), sleep_quality (-0.008), offensive_performance (-0.009), HIR (-0.015), and sleep_duration (-0.019). These features show minimal relationship with injury, suggesting they are less likely to be useful in predicting injury occurrences.

We also checked important feature-to-feature correlations:

Monotony and Strain: A strong positive correlation (0.8999) indicates that as monotony increases, so does reported strain.

Weekly Load and Daily Load: A moderate positive correlation (0.5493) suggests that athletes

with higher weekly loads tend to have higher daily loads.

CTL42 and CTL28: A very strong positive correlation (0.9416) suggests these are closely related measures.

Sleep Quality and Mood: A moderate positive correlation (0.4095) indicates that better sleep quality is associated with a better mood.

Offensive and Defensive Performance: A strong positive correlation (0.6556) suggests they tend to improve or decline together.

Weekly Load and Stress: A moderate negative correlation (-0.1649) suggests that higher weekly loads might be associated with lower reported stress levels.

HIR and Daily Load: A moderate positive correlation (0.3404) indicates that higher daily loads may involve more high-intensity running.

Stress and Mood: A strong positive correlation (0.5958) indicates that stress and mood tend to move in the same direction.

Sleep Quality and Stress: A moderate positive correlation (0.4074) suggests that higher stress levels may be associated with poorer sleep quality.

Readiness and Sleep Quality: A moderate positive correlation (0.1714) suggests that higher readiness might be associated with better sleep quality.

ACWR and CTL42/CTL28: Moderate negative correlations (-0.4073 and -0.3749, respectively) indicate an inverse relationship between the acute-to-chronic workload ratio and CTL42/CTL28.

Injury Timestamps (Injury_ts) and HIR: A weak negative correlation (-0.0205) implies a slight inverse relationship between injury timestamps and the Health Impact Ratio.

3.4.2 Pairplot Visualization

We used Seaborn's pair plot to visualize important features with the following code:

```
1
2 important_columns = ['monotony', 'weekly_load', 'strain', 'daily_load', 'ctl42'
      , 'ctl28', 'defensive_performance', 'injury']
3 sns.pairplot(df[important_columns])
4 plt.show()
5
6 slightly_important_columns = ['team_performance', 'atl', 'stress', 'illness', '
    Total_distance', 'readiness', 'acwr', 'injury_ts', 'mood', 'injury']
7 sns.pairplot(df[slightly_important_columns])
8 plt.show()
```

The analysis of the plots revealed that individuals with longer sleep durations tend to experience lower levels of soreness. However, no clear patterns or correlations were found between sleep duration, injury, and soreness. The scatter plots indicated that injuries were scattered across different levels of these features, suggesting that they may not strongly predict injury occurrence.

Additionally, while correlations were observed between variables like readiness and illness, readiness and fatigue, and strain and injury, the scatter plots indicated that these variables do not strongly predict injury risk. Mood levels were also found to have minimal impact on injury rates. The analysis did not explicitly discuss the relationships between defensive and offensive performance with other factors.

3.4.3 Boxplot Visualization

We applied Seaborn's boxplot visualization on numerical features to identify outliers using the following code:

```
numerical_columns = ['monotony', 'weekly_load', 'strain', 'daily_load', 'ctl42'
    , 'ctl28', 'defensive_performance', 'team_performance','atl', 'stress', '
    illness', 'Total_distance', 'readiness', 'acwr', 'injury_ts', 'mood','
    sleep_quality', 'offensive_performance', 'HIR']

plt.figure(figsize=(12, 8))
sns.boxplot(data=df[numerical_columns], orient='h').set(title='Box Plot of
    Numerical Columns with Outliers', xlabel='Values', ylabel='Features')
plt.show()
```

Upon analyzing the dataset, we noticed the presence of notable outliers in various columns, indicating the potential existence of anomalies or exceptional circumstances within the measured features. Specifically, in the atl column, a few data points were observed at higher values, signifying deviations from the central distribution. These outliers may represent exceptional situations or unusual occurrences within the dataset. Similarly, the ctl28 column also exhibited several outliers at higher values, suggesting the presence of unique circumstances or exceptional conditions within the dataset. These outliers warrant further investigation to understand their underlying causes and assess their impact on the overall analysis and conclusions drawn from the data.

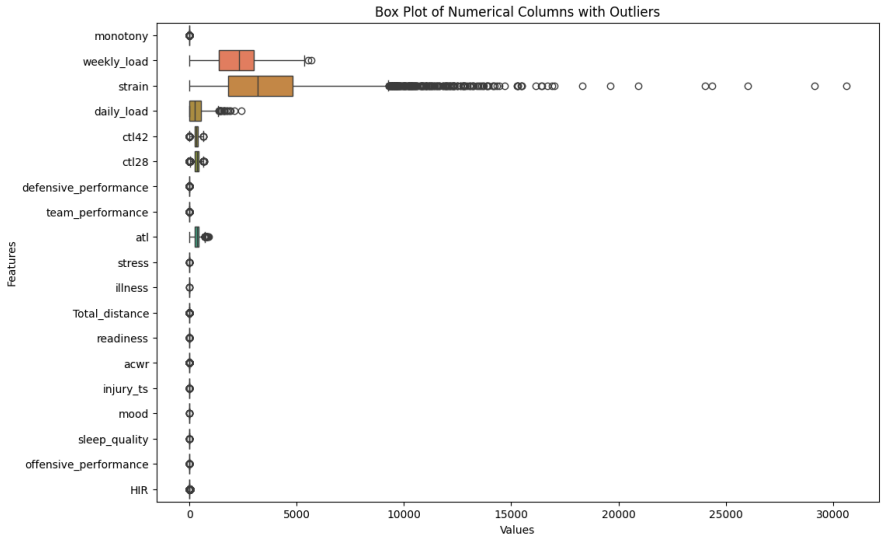

Figure 3.2: The box plot shows the distribution and outliers of various numerical features, with the boxes representing the interquartile range and whiskers extending to 1.5 times the IQR.

In addition to the atl and ctl28 columns, the ctdl2 column also displays outliers across a wide range of values. These outliers might indicate specific events or exceptional circumstances that significantly influence the overall trend of this feature. Interestingly, we observed a single outlier at a higher value in the daily_load column, which could be due to unusual circumstances or a potential measurement error.

Moreover, both the strain and weekly_load columns reveal numerous outliers across a wide range of high values. These outliers could represent critical stress levels or exceptional conditions experienced by the individuals in the dataset. The presence of these outliers highlight the importance of using robust data analysis techniques. These techniques can help us account for and accurately interpret these extreme observations, ensuring the reliability of our subsequent analyses and findings.

3.4.4 Time Series Analysis

Considering the temporal nature of our dataset, we carried out a variety of time series analyses to thoroughly evaluate our data. These analyses included seasonal analysis, autocorrelation, and partial autocorrelation. Seasonal analysis allowed us to understand the patterns or trends that repeat over specific intervals in our data. Autocorrelation analysis helped us to determine the relationship between a variable's current value and its past values. Meanwhile, the partial autocorrelation analysis aided in determining the direct effect of past values on the current value of a variable. By performing these time series analyses, we were able to gain deeper insights into the temporal structure of our data, which is crucial for accurate forecasting and modeling.

Seasonal Analysis

To identify recurring patterns in the concentration levels of various performance and well-being metrics, we performed a seasonal analysis on a monthly basis. We plotted the average concentration per month for each column using the following code:

```
fig, ax = plt.subplots(figsize=(20, len(important_columns)*4), nrows=len(
    important_columns))

for idx, var in enumerate(important_columns):
    data_monthly[[var]].groupby(data_monthly.index.month).mean().plot(ax=ax[idx
    ], color='RosyBrown', title=f"{var.capitalize()} Concentration Seasonality
    - Monthly", ylabel="Concentration", xlabel="Month of the Year").grid(axis='
    y')

plt.tight_layout()
plt.show()
```

Upon analyzing monthly seasonality across various performance and well-being metrics, we observed distinct patterns in concentration levels throughout the year. For parameters such as sleep quality, defensive performance, sleep duration, soreness, illness, readiness, fatigue, offensive performance, mood, daily load, strain, and ACWR, we noticed cyclic trends. These trends indicate that concentration levels fluctuate in a recurring manner over time. Recognizing these patterns helps us understand the temporal dynamics of these metrics and can inform strategies to enhance performance and well-being based on these cyclical patterns.

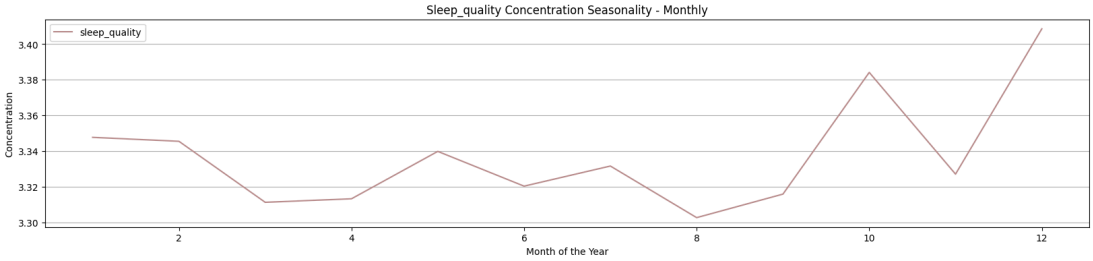

Figure 3.3: The plot illustrates the monthly seasonality of sleep quality concentration throughout the year. The data shows fluctuations, with a noticeable dip around mid-year and a peak in December.

We observed patterns of peaks and troughs at specific months in the data, suggesting a seasonal influence on individual metrics. For instance, we noticed that variables like readiness, fatigue, offensive performance, mood, daily load, strain, and ACWR peak in concentration around the middle of the year and decline rapidly after October. This suggests that performance and workload might fluctuate seasonally.

On the other hand, metrics such as sleep quality and illness showed different seasonal patterns, with concentrations varying across different months. These variations could be influenced by external factors such as weather changes, training schedules, or competition calendars.

By understanding these seasonal trends, we can gain valuable insights that could help in optimizing performance, managing workload, and effectively planning interventions throughout the year. Recognizing when certain metrics are likely to peak or drop can guide us in developing strategies to enhance performance and well-being.

Autocorrelation

Autocorrelation measures how a time series is correlated with its own past values. We applied autocorrelation analysis using two different approaches:

Lag Plots: We applied autocorrelation analysis using two different visualization approaches - Lag Plots and Advanced Lag Plots with Regression Lines. Lag plots helped us visualize the relationship between time-lagged values for each feature, providing a graphical representation of how each observation relates to its past values. To further understand the linear relationship between these lagged values, we used Advanced Lag Plots with Regression Lines. The addition of regression lines to the lag plots allowed us to quantify the strength and direction of these relationships. These approaches offered us valuable insights into the time-dependent structure of our data, aiding in the creation of more accurate predictive models. The following code was used for these analyses:

```
1
2 def create_lag_plots(data, lags=6):
3     fig, axs = plt.subplots(len(data.columns), lags, figsize=(40, 60))
4     [[axs[i][j-1].scatter(data.shift(j)[col], data[col], alpha=0.5), axs[i][j
      -1].set(xlabel=f"{col} (lagged {j})", ylabel=col, title=f"Time Lagged Plot
      of {col} (Lag {j})")] for i, col in enumerate(data.columns) for j in range
      (1, lags+1)]
5     plt.tight_layout()
```

```
    plt.show()

create_lag_plots(df[important_columns], lags=6)

def create_lag_plots_adv(data, lags=5):
    plt.figure(figsize=(44, 60))
    [sns.regplot(x=data.shift(j)[col], y=data[col], scatter_kws={"s": 10},
    line_kws={"color": "red"}).set(xlabel=f"{col} (lagged {j})", ylabel=col,
    title=f"Lag Plot of {col} (Lag {j})") for i, col in enumerate(data.columns)
     for j in range(1, lags+1)]
    plt.tight_layout()
    plt.show()

create_lag_plots_adv(df[important_columns], lags=5)
```

We have used lag plots to visualize the relationship between the time-lagged values of each feature. This was key in understanding the temporal dependencies within our data. For instance, these plots could demonstrate how past values of specific features like weekly_load or strain might influence current values. This information proved beneficial when creating predictive models, as it allowed us to consider the impact of past observations on current values. By understanding these relationships, we were able to build more accurate and effective predictive models.

Partial Autocorrelation

Partial autocorrelation measures the correlation between observations of a time series and its lagged values, while controlling for the values of the time series at all shorter lags. This analysis provides insights into the direct relationship between an observation and its lagged value, excluding the influence of other lagged values. To further analyze our data, we utilized partial autocorrelation using the following code:

```

[plot_pacf(df[col], lags=30, title=f'Partial Autocorrelation Plot of {col}')
    for col in important_columns]
```

By plotting partial autocorrelations, we were able to gain a deeper understanding of the temporal structure of each feature. We identified significant lags that directly influence current values, which is crucial information when dealing with time series data. This understanding of how past values impact current ones is valuable in building accurate time series algorithms. It allows us to make data-driven decisions based on patterns observed in historical data, ultimately enhancing the reliability and effectiveness of our predictive models.

3.5 Feature Engineering

After thoroughly analyzing our data, we used various feature engineering techniques to improve our data for our algorithms. We created new features from existing ones, standardized the range of our features, and compiled broader trends from our data. We also organized our data to highlight patterns, analyzed the importance of each feature, selected the most relevant features, and handled redundant features. Each of these steps was carefully carried out to enhance the quality of our data and improve the predictive power of our models.

3.5.1 Feature Extraction

To enhance our dataset, we have created several new features. One of the first additions was the historical injury count for each player. This feature provides insight into each player's injury history, which can significantly predict future injuries. By tracking the number of previous injuries, we can consider the player's injury susceptibility, which can greatly improve the accuracy of our injury prediction models.

```
df['historical_injury_count'] = df.groupby('player_names')['injury'].apply(np.
    cumsum)
```

Next, we introduced interaction features to capture the relationships between various performance metrics. These interaction features are critical as they help us understand how different combinations of factors influence the outcome. By analyzing how these metrics interact with each other, we can gain a deeper insight into the data. This allows us to identify complex relationships and patterns that might not be evident when looking at individual metrics alone, ultimately enhancing the effectiveness of our predictive models.

```
interactions = [('daily_load', 'readiness'), ('sleep_duration', 'sleep_quality'
    ),
                ('daily_load', 'fatigue'), ('strain', 'readiness'), ('stress',
    'readiness'),
                ('team_performance', 'readiness'), ('offensive_performance', '
    readiness'),
                ('defensive_performance', 'readiness'), ('defensive_performance
    ', 'weekly_load')]

for col1, col2 in interactions:
    df[f'{col1}_{col2}_interaction'] = df[col1] * df[col2]
```

3.5.2 Feature Scaling

To ensure all numerical features were on the same scale and improve model performance, we applied Min-Max scaling. This normalization technique is an essential step in feature engineering as it brings all features to the same scale, reducing the likelihood of one or more features dominating the model due to their larger numerical ranges. This is particularly beneficial for algorithms that are sensitive to the scale of the data, helping to improve the accuracy and reliability of our machine learning models.

```

df[['acwr', 'weekly_load', 'stress']] = MinMaxScaler().fit_transform(df[['acwr'
    , 'weekly_load', 'stress']])
```

Feature scaling is a crucial step in the data preprocessing phase. It ensures that all features contribute equally to the model training process, preventing features with larger scales from dominating those with smaller scales. This is particularly important for many machine learning algorithms, especially those that rely on distance calculations, such as K-Nearest Neighbors, and Support Vector Machine. By bringing all features to the same scale, feature scaling helps improve the accuracy and performance of these algorithms.

3.5.3 Aggregated Metrics

We introduced aggregated metrics to capture long-term trends and smooth out short-term fluctuations in our data. These metrics consolidate data over a specific period, providing a

summary view that can highlight underlying patterns and trends. This aggregation can help uncover insights that might not be immediately visible in the raw data, allowing us to better understand the broader context and make more informed predictions. Aggregated metrics are particularly useful in time series analysis, where understanding long-term trends can be crucial for accurate forecasting.

```
df['acwr_rolling_avg'] = df['acwr'].rolling(window=7).mean()
df['weekly_load_ema'] = df['weekly_load'].ewm(alpha=0.5).mean()
df['stress_rolling_sum'] = df['stress'].rolling(window=14).sum()
```

We used rolling averages, exponential moving averages, and rolling sums to see how these metrics change over time. These aggregated metrics helped us spot trends and potential unusual data points. They are especially useful when analyzing time series data, as they can help identify patterns over time. By smoothing out short-term changes, these metrics allow us to focus on long-term trends, which can improve the accuracy of our predictive models.

3.5.4 Sorting Data

To maintain the temporal order of the data, we sorted the dataset by player names and dates. This step is crucial for time series analysis as it ensures that the model trains on sequential data. By training our algorithms on data in the correct sequence of events, we can create models that accurately learn from past data to predict future outcomes. This step is particularly important for our dataset, given its temporal nature, and greatly contributes to the accuracy and reliability of our predictive models.

```
sorted_df = df.sort_values(by=['player_names', 'Date'])
```

We sorted the data by player names and dates to preserve the chronological order of the observations. This is critical for time-dependent analyses. It ensures that we maintain the temporal relationships within the data, allowing our algorithms to accurately capture and learn from the sequential patterns and trends. By keeping the data in order, the models we develop are better equipped to make precise predictions based on these learned patterns.

3.5.5 Feature Importance Analysis and Feature Selection

Identifying the most relevant features is key to improving model performance and interpretability. We used several methods, including feature correlation, Random Forest Classifier, and Gradient Boosting Classifier, to calculate feature importance. This comprehensive approach helped us understand which features were most informative and should be prioritized during model training. By focusing on the most relevant features, we can create more accurate and efficient models, reducing noise and unnecessary complexity in our data.

First, we examined the correlation of each feature with the target variable 'injury'. This step is crucial as it helps in identifying features that have a strong linear relationship with the outcome. By examining these correlations, we can identify the features that are most likely to influence the target variable. These features are often the most important for predictive modeling, and understanding their relationships with the target variable can provide valuable insights for model training.

```
correlation_matrix_injury = df.corr()[['injury']].sort_values(by='injury',
    ascending=False)

for col in correlation_matrix_injury.index:
```

```
    correlation = correlation_matrix_injury.loc[col, 'injury']
    print(f"Correlation with 'injury' for column '{col}': {correlation}")
```

Next, we calculated feature importance using two ensemble methods: Random Forest and Gradient Boosting. These algorithms help determine the importance of each feature by evaluating how much each feature contributes to reducing the prediction error. By understanding the significance of each feature in the context of these models, we can identify the most influential features in our dataset. This information is crucial for building effective predictive models, as it allows us to focus on the features that have the greatest impact on our predictions.

```
def calculate_feature_importance(df, model):
    X, y = df.drop('injury', axis=1), df['injury']
    return pd.Series(model.fit(X, y).feature_importances_, index=X.columns).
    sort_values(ascending=False)

rf_feature_importances = calculate_feature_importance(df,
    RandomForestClassifier())
gb_feature_importances = calculate_feature_importance(df,
    GradientBoostingClassifier())
```

After calculating the feature importance using both the Random Forest and Gradient Boosting algorithms, we combined these scores and calculated an average importance score for each feature. This step helped to stabilize the importance rankings and made sure that the selected features were consistently important across different methods. By doing this, we identified a set of features that were important to both algorithms, making our feature selection process more reliable.

```
def select_top_features(feature_importances_list, correlation_matrix,
    correlation_threshold=0.005, num_features_to_select=27):
    selected_features_combined = pd.concat(feature_importances_list, axis=1,
    keys=['RF', 'GB']).mean(axis=1).sort_values(ascending=False)
    correlated_features = [col for col, corr in correlation_matrix['injury'].
    drop('injury').items() if abs(corr) >= correlation_threshold]
    return [feature for feature in selected_features_combined.index if feature
    in correlated_features][:num_features_to_select]

correlation_matrix = df.corr()
final_selected_features = pd.DataFrame([select_top_features([
    rf_feature_importances, gb_feature_importances], correlation_matrix) for _
    in range(5)]).mode().iloc[0].tolist()
```

The final selected features, which were found to be most important and relevant across all iterations, include: ‘’historical injury count’, ’ctl42’, ’defensive performance weekly interaction’, ’ctl28’, ’strain readiness interaction’, ’acwr rolling avg’, ’offensive performance’, ’atl’, ’strain’, ’acwr’, ’Top speed’, ’load fatigue interaction’, ’team performance’, ’monotony’, ’Average running speed’, ’weekly load ema’, ’stress rolling sum’, ’fatigue’, ’weekly load’, ’team performance readiness interaction’, ’defensive performance readiness interaction’, ’HIR’, ’offensive performance readiness interaction’, ’sleep readiness interaction’, ’daily load’, ’mood’, ’load readiness interaction’‘.

By focusing on these selected features, we can ensure that our algorithms are trained on the most informative and relevant data. This leads to better predictive performance and more reliable outcomes. This careful selection process not only reduces noise in our data, but also improves the interpretability of our models. By focusing on the most impactful features, we can create more effective predictive algorithms, enhancing the overall accuracy of our models.

3.5.6 Feature Redundancy

We applied feature correlation analysis to identify feature pairs with relatively high correlation coefficients that could potentially mislead algorithms in injury prediction. Our observations are as follows:

Sleep Quality and Sleep Readiness Interaction: There is a strong positive correlation of 0.917 between sleep quality and sleep readiness interaction, indicating that higher sleep readiness tends to coincide with better sleep quality.

Stress and Mood: Stress and mood have a strong positive correlation of 0.616, suggesting that higher levels of stress are associated with poorer mood.

Stress and Stress Readiness Interaction: Stress and stress readiness interaction have a very strong positive correlation of 0.742, indicating that higher stress readiness is associated with higher stress levels.

Weekly Load and ACWR (Acute: Chronic Workload Ratio) Rolling Average: There is a positive correlation of 0.881 between weekly load and ACWR rolling average, suggesting that as weekly load increases, the ACWR tends to increase as well.

Monotony and ATL (Acute Training Load): Monotony and ATL have a moderate positive correlation of 0.648, indicating that higher levels of monotony tend to coincide with higher acute training loads.

Strain and Strain Readiness Interaction: Strain and strain readiness interaction have a very strong positive correlation of 0.938, indicating that higher strain readiness tends to coincide with higher levels of strain.

Load Readiness Interaction and Daily Load: Load readiness interaction and daily load have a very strong positive correlation of 0.963, indicating that higher load readiness tends to coincide with higher daily loads.

To handle feature redundancy, we identified and removed highly correlated features (correlation coefficient > 0.9). Redundant features can mislead the model and reduce performance, so it is crucial to eliminate them:

```
1
2 correlation_threshold = 0.9
3 correlation_matrix = df.corr().abs()
4 redundant_features = [correlation_matrix.columns[j] for i, j in zip(*np.where(
      np.triu(correlation_matrix > correlation_threshold, k=1)))]
5 df = df.drop(columns=redundant_features)
```

3.5.7 Final Features

After extensive feature engineering and redundancy removal, we selected the following features to retain in our dataset: 'daily load', 'fatigue', 'mood', 'readiness', 'sleep duration', 'sleep quality', 'soreness', 'stress', 'injury ts', 'weekly load', 'Team', 'Total distance', 'Average running speed', 'Top

speed', 'HIR', 'acwr', 'atl', 'ctl28', 'monotony', 'strain', 'team performance', 'offensive performance', 'defensive performance', 'illness', 'historical injury count', 'stress readiness interaction', 'team performance readiness interaction', 'offensive performance readiness interaction', 'defensive performance readiness interaction', 'acwr rolling avg', 'stress rolling sum', 'injury', and 'Player name'. Table 3.1. provides an overview of all the selected features name, descriptions, and metrics.

We selected these particular features based on their predictive importance and relevance to the target variable. These features provide a robust foundation for building accurate and effective algorithms. By focusing on these selected features, we ensure that our algorithms are trained on the most relevant and informative data. This approach leads to improved predictive performance and more reliable outcomes. This careful selection process helps reduce noise in our data and enhances both the interpretability and effectiveness of our predictive algorithms.

Parameter	Description	Metric
Daily Load	Sum of sRPE per single day	Numeric
Fatigue	Feeling of constant exhaustion	Numeric
Mood	Players emotional state	Numeric
Readiness	Players readiness for a training session or game	Numeric
Sleep Duration	Duration of sleep	Numeric
Sleep Quality	Quality of Sleep	Numeric
Soreness	Level of soreness	Numeric
Stress	Current level of stress	Numeric
Injury TS	Binary indication of injury with 1 suggesting an injury	Numeric
Weekly Load	Sum of sRPR over seven days	Numeric
Team	Letter describing which team a given player belongs to	Numeric
Total Distance	Total distance covered during the session	Numeric
Average Running Speed	The average running speed of the session	Numeric
Top Speed	Top speed of the session	Numeric
HIR	Number of high intensity runs throughout the session	Numeric
ACWR	Acute load related to the chronic load	Numeric
Atl	Acute training load	Numeric
Ctl28	Chronic training load of the last 28 days	Numeric
Monotony	ATL divided by standard deviation	Numeric
Strain	Training stress over the last 7 days	Numeric
Team Performance	Performance of the team	Numeric
Offensive Performance	Offensive performance of the team	Numeric
Defensive Performance	Defensive performance of the team	Numeric
Illness	Level of illness	Numeric
Historical Injury Count	Sum of occurrence of injury per player	Numeric
Stress Readiness Interaction	Combination of stress and readiness	Numeric
Team Performance Readiness Interaction	Combination of team performance and readiness	Numeric
Offensive Team Performance Readiness Interaction	Combination of offensive team performance and readiness	Numeric
Defensive Team Performance Readiness Interaction	Combination of defensive team performance and readiness	Numeric
ACWR Rolling Average	Average of acute chronic workload ratio rolling	Numeric
Stress Rolling Sum	Sum of stress rolling	Numeric
Injury	Occurrence of injury	Numeric
Player Name	Unique ID of the soccer players	Object ID

Table 3.1: The table shows all the selected features after feature engineering.

3.6 Final Dataset

After comprehensive preprocessing and feature engineering, we compiled our final dataset, which consists of a total of 8,582 rows. Out of these, 8,526 rows represent non-injury cases (labeled as 0),

and only 56 rows represent injury cases (labeled as 1). This reveals a significant class imbalance in our dataset, with a much larger number of non-injury cases compared to injury cases. Such class imbalance can pose a challenge for model training as it could lead to a model that is biased towards predicting the majority class. Therefore, addressing this imbalance will be an important step in our model development process to ensure accurate and reliable predictions.

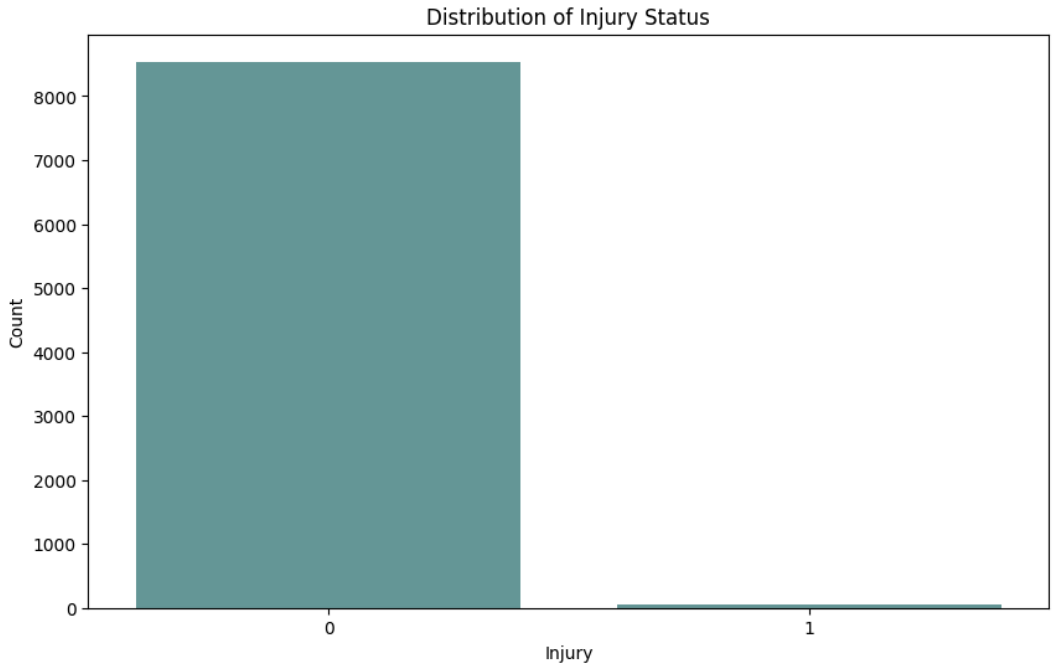

Figure 3.4: The bar chart displays the distribution of injury status, with the vast majority (over 8000) of instances having no injury (0) and a very small number indicating injury (1).

3.6.1 Multiple Datasets

To optimize our results and address potential biases, we created multiple versions of the dataset:

Original Version (Full Dataset): In the original version, which we referred to as the full dataset throughout our thesis, we retained the dataset exactly as it was initially prepared, with all its features and instances. This version serves as a baseline for comparison, allowing us to assess the impact of any changes we make later on. It includes all the information we collected, providing a comprehensive view of the data for our analyses.

Team A Version (Single Team Dataset): In several parts of our thesis, we created and referred to a version known as the single team dataset. In this version, we included only players from Team A to create a dataset. This approach allowed us to examine injury patterns and evaluate model performance within the specific context of a single team. By focusing on a specific team, we were able to gain valuable insights into the unique dynamics of injuries within that team, providing a more detailed and nuanced understanding of the factors at play.

Filtered Version (Only Injured Players Dataset): The filtered version, which we referred to as the only injured players dataset, was created by removing rows where players did not have any injury data. This filtering process resulted in a dataset that consisted of 3,134 non-injury rows and 56 injury rows. The purpose of this version is to focus on players with a history of injuries. By doing so, we aimed to potentially enhance the model's ability to recognize patterns related to injuries. This concentrated view on injured players could provide more specific insights into the factors contributing to injuries.

By creating these different versions of the dataset, we aimed to thoroughly explore various aspects of the data and enhance the robustness of our algorithms. Each version offers a unique perspective, providing a different lens through which to examine the data. This approach allowed us to better understand the factors influencing injury prediction. It also helped us identify the most effective strategies for handling imbalanced data and improving the accuracy of our predictions. This multi-faceted approach to data analysis contributes to a more comprehensive and nuanced understanding of the data, ultimately enhancing the quality of our predictive models.

3.7 Injury Prediction

Once we had our datasets prepared, we implemented various machine learning and deep learning algorithms to predict injuries. To improve the accuracy of our models, we applied hyperparameter tuning to optimize the parameters of our algorithms. We also used sliding windows, a technique often used in time series analysis, to ensure our models could learn from sequential data. Additionally, we employed techniques to handle class imbalance in our data, ensuring our models could accurately predict both injury and non-injury cases. By applying these strategies, we aimed to create robust, accurate, and reliable injury prediction models.

3.7.1 Sliding Windows

Considering the temporal nature of our dataset, we used sliding windows to train our algorithms. This approach helps capture temporal dependencies and patterns in the data, which is crucial for time series analysis. We experimented with different window sizes, such as 2, 4, 8, 16, and 32, to determine the optimal window size that provides the most accurate results. The following code snippet illustrates how we implemented sliding windows:

```
def create_player_sequences(df, player, sequence_length=7):
    player_data = df[df['Player_name'] == player].drop(['Player_name'], axis=1)
    .values
    return np.array([player_data[i:i + sequence_length] for i in range(len(
    player_data) - sequence_length)])

def create_team_sequence(df, sequence_length):
    return np.concatenate([create_player_sequences(df, player, sequence_length)
     for player in df['Player_name'].unique()], axis=0)

def preprocess(inputWindow):
    df = pd.read_csv(dataPath)
    X, y = create_team_sequence(df, inputWindow)
    return X.astype(np.float32), y.astype(np.float32)
```

By using sliding windows, we made our algorithms better at capturing how player data changes over time. This approach considers the sequence of the data, which is very important for time series

analysis. By doing so, our algorithms could better understand and learn from the data patterns, leading to more accurate predictions. This method was key in improving our predictive models.

3.7.2 Class Imbalance Handling

To tackle the issue of class imbalance in our data, where injury cases (the minority class) are significantly outnumbered by non-injury cases (the majority class), we implemented several techniques. These included oversampling, undersampling, SMOTE, and ADASYN sampling methods.

Oversampling: Oversampling is a technique we applied to balance the class distribution in our data. This method involves increasing the number of instances in the minority class, in this case, the positive class in our training data. It does this by randomly replicating instances from the minority class to match the number of instances in the majority class. By using oversampling, we were able to create a more balanced dataset, reducing the bias towards the majority class and improving the performance of our predictive models.

Undersampling: Undersampling is another technique we used to balance the class distribution in our data. In contrast to oversampling, undersampling involves reducing the number of instances in the majority class, in our case, the negative class in the training data. It achieves balance by randomly removing instances from the majority class to match the number of instances in the minority class. By using undersampling, we were able to create a more balanced dataset, reducing the bias towards the majority class and improving the performance of our predictive models.

SMOTE: The Synthetic Minority Over-sampling Technique is another method we used to balance the class distribution in our data. Unlike traditional oversampling, which simply replicates instances from the minority class, SMOTE generates synthetic samples. It does this by creating synthetic samples along the line segments joining the k nearest neighbors of minority class samples. This method allows us to increase the number of instances in the minority class without simply replicating existing data, thereby enhancing the diversity of the training data and improving the performance of our predictive models.

ADASYN: The Adaptive Synthetic Sampling Approach for Imbalanced Learning is another technique we used to balance our class distribution. Similar to SMOTE, ADASYN generates synthetic samples for the minority class. However, ADASYN focuses more on regions where the class imbalance is most severe, generating more synthetic samples in those specific areas. By doing so, ADASYN ensures that the model pays more attention to these harder-to-learn instances, leading to improved performance of our predictive models.

The following code snippet demonstrates the implementation of these techniques:

```
1 def resample_data(X_train, y_train, sampling_ratio, oversample_mode):
2     func_map = {'oversample': RandomOverSampler, 'undersample':
    RandomUnderSampler, 'smote': SMOTE, 'adasyn': ADASYN}
3     resampler = func_map[oversample_mode](sampling_strategy=sampling_ratio,
    random_state=42)
4     X_resampled, y_resampled = resampler.fit_resample(X_train.reshape(X_train.
    shape[0], -1), y_train)
5     precentageOfZeroesInDataset = (y_resampled == 0).sum() / len(y_resampled)
```

```
    return X_resampled.reshape(X_resampled.shape[0], X_train.shape[1], X_train.shape[2]), y_resampled, precentageOfZeroesInDataset
```

By implementing these techniques, we effectively addressed the class imbalance in our data. As a result, our algorithms were able to make more accurate and reliable predictions. Balancing the classes in our data ensured that both the majority and minority classes were adequately represented in our training data, leading to more robust and unbiased predictive models. This was a crucial step in improving the overall performance of our models.

3.7.3 Models Implementation

We deployed a diverse array of machine learning algorithms, leveraging the capabilities of the 'sklearn' library, to predict injuries. These algorithms include Decision Tree, K-Nearest Neighbors, Support Vector Machine, Logistic Regression, Naive Bayes, Random Forest, and XGBoost. Each model offers unique strengths and characteristics that contribute to the overall predictive performance. Additionally, recognizing the potential benefits of deep learning in capturing temporal dependencies within the data, we also implemented a Long Short-Term Memory (LSTM) model. LSTM is a type of Recurrent Neural Network (RNN) that excels at processing sequences of data, making it particularly well-suited for time series analysis and sequential data like ours.

By using both traditional machine learning algorithms and deep learning architectures, we aimed to make our injury predictions more accurate and reliable. Combining these methods let us take advantage of the benefits of different modeling approaches. Traditional machine learning algorithms are simple and easy to interpret, while deep learning architectures are powerful and good at recognizing patterns. This mix of methods improved our predictive modeling approach, helping us create a system that can make reliable and accurate injury predictions.

3.7.4 Models Training

In our model training phase, we partitioned our dataset into training and testing sets, allocating 80% and 70% of the data for training, respectively, and reserving the remaining portion for testing. This split was conducted to ensure an adequate amount of data for training while maintaining a sufficient portion for evaluating model performance. We experimented with different sample ratios, using both 0.2 and 0.3 ratios to explore their impact on model accuracy.

For traditional machine learning algorithms such as Decision Tree, we used the Decision Tree Classifier from the 'sklearn' library. The training process involved preprocessing the data, splitting it into training and testing sets, and then fitting the model to the training data. The following code snippet is a sample of the machine learning model (Decision Tree model) implementation:

```
def decision_tree_classification(test_size, oversample_mode, sampling_ratio):
    X_scaled = TimeSeriesScalerMinMax().fit_transform(X)
    X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=test_size, random_state=42, stratify=y)
    precentageOfZeroesInDataset = (y_train == 0).sum() / len(y_train)
    X_train, y_train, precentageOfZeroesInDataset = sample_mode(X_train, y_train, sampling_ratio, oversample_mode)
    model = DecisionTreeClassifier().fit(X_train.reshape(X_train.shape[0], -1), y_train)
    y_pred = model.predict(X_test.reshape(X_test.shape[0], -1))
    confInjuries = confusion_matrix_only_injuries(y_test, y_pred)
```

```
    return confInjuries, precentageOfZeroesInDataset
```

Furthermore, we ventured into deep learning territory by implementing an LSTM model using TensorFlow and Keras. LSTM is particularly well-suited for handling sequential data, making it an ideal candidate for our time series dataset. The LSTM model architecture consisted of multiple LSTM layers followed by dropout layers to prevent overfitting and a final dense layer with a sigmoid activation function. We trained the LSTM model using various combinations of hyperparameters, including input window sizes, test sizes, oversampling methods, and sampling ratios. The following code snippet demonstrates the implementation and training procedure for the LSTM model.

```

def create_lstm_model(input_shape):
    model = Sequential([LSTM(64, input_shape=input_shape, return_sequences=True
    ),
                        Dropout(0.2),
                        LSTM(32),
                        Dropout(0.2),
                        Dense(1, activation='sigmoid')])
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['
    accuracy'])
    return model
```

Throughout the training process, we evaluated each model's performance using several metrics. By systematically exploring different hyperparameter combinations and model architectures, we aimed to identify the most effective strategies for predicting injuries in our dataset.

3.7.5 Models Performance Evaluation

After training our machine learning and deep learning algorithms, we proceeded to evaluate their performance using a comprehensive set of metrics. For traditional machine learning algorithms like Decision Tree, we assessed accuracy, F1-score, precision, and recall to gauge their predictive capability. These metrics provided insights into the model's overall accuracy, and its ability to balance between correctly identifying positive cases (injuries) and avoiding false positives.

In contrast, we utilized the same metrics to evaluate its performance on our sequential data. Additionally, we used techniques such as time series cross-validation to ensure robust evaluation, accounting for the temporal nature of our dataset. By comparing the performance across different hyperparameter combinations and model architectures, we were able to identify the most effective strategies for injury prediction.

Moreover, we ranked the algorithms based on their F1 scores to determine the top performers, providing valuable insights into which configurations yielded the best predictive performance. This rigorous evaluation process allowed us to select the most promising algorithms for deployment in real-world scenarios, where accurate injury prediction is of utmost importance for athlete well-being and performance optimization.

3.7.6 Hyperparameter Tuning

Hyperparameter tuning is key to improving the performance of machine learning algorithms. It involves finding the best set of parameters for a given algorithm to improve its accuracy. In the code below for the Decision Tree Classifier, we used a method called Grid Search Cross Validation

for hyperparameter tuning. This method helps us find the best parameters for our model by testing different combinations. By fine-tuning the parameters, we were able to improve the performance of our Decision Tree Classifier and make our injury predictions more accurate.

```
param_grid = {
        'criterion': ['gini', 'entropy'],
        'max_depth': [None, 10, 20, 30],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4],
        'max_features': ['auto', 'sqrt', 'log2']
    }
```

The process starts by defining a grid of hyperparameters and their respective values, such as the criterion for splitting ('gini' or 'entropy'), maximum depth of the tree, minimum samples required to split an internal node, minimum samples required to be at a leaf node, and the maximum number of features to consider for the best split. Grid Search Cross Validation then exhaustively searches through all possible combinations of these hyperparameters to identify the optimal configuration based on a specified evaluation metric, typically accuracy, F1-score, or another relevant metric.

Once the search is complete, the best set of hyperparameters is selected based on the performance of the model on cross-validated data. This tuned model is then evaluated on the test set to assess its performance metrics, including accuracy, F1 score, precision, and recall.

Hyperparameter tuning ensures that our algorithms are fine-tuned to extract the maximum predictive power from the data, leading to more robust and reliable predictions. By systematically exploring different hyperparameter configurations, we can uncover the optimal settings that yield superior performance, enhancing the overall effectiveness of our machine learning algorithms.

3.8 Chapter Summary

This chapter offers a comprehensive overview of the methodology and implementation of our machine learning framework to predict injuries among Norwegian women's soccer players. We began with a proposed plan, followed by a discussion on data import using Python libraries. This led to the crucial step of data preprocessing, setting the stage for accurate and efficient analysis. The exploratory data analysis (EDA) that followed involved creating various plots to dig deeper into our data and unearth key insights.

In the feature engineering section, we explored different aspects, including feature extraction, feature scaling, and handling feature redundancy, among others. The process of creating our final dataset and the two additional datasets used for various experiments was also described. The chapter concluded with an explanation of the sliding window approach, methods for handling class imbalances, model implementation, model training, and model evaluation. We also touched upon the process of hyperparameter tuning. Each of these steps collectively contributed to the robustness of our machine learning framework.

In the next chapter, we will present the results of seven experiments that were conducted to answer specific preliminary, and sub-questions. These experiments were carried out using the methods and implementation process described earlier.

Chapter 4

Experiments and Results

In the previous chapter, we provided a detailed account of our implementation methodology, focusing on the preprocessing steps, feature engineering, and the application of various machine learning and deep learning algorithms. This laid the foundation for our predictive analysis of soccer player injuries.

In this chapter, we present the results obtained from our predictive algorithms. We will analyze the performance of each algorithm, comparing metrics such as accuracy, precision, recall, and F1-score to determine their effectiveness. Additionally, we will discuss the outcomes of hyperparameter tuning and the impact of handling class imbalance on our algorithm's performance. By interpreting these results, we aim to validate our approach and highlight the practical implications of our findings in the context of sports analytics. This chapter serves to bridge our methodological framework with its empirical evaluation, providing insights into the predictive power and reliability of our injury prediction algorithms.

4.1 Experiment 1: Identify Key Risk Factors for Injury

The correlation analysis conducted in our study provided significant insights into the factors associated with injury occurrence. We have discussed the highly correlated features with injury, as well as the correlation between different features, in section 3.4.1. Here is a detailed summary of our observations:

Highly Correlated Features: The Figure 4.1. shows that features such as illness, weekly_load, ctl28, atl, ctl42, monotony, strain, and acwr exhibit relatively high correlations with the injury column. This suggests that higher values of these features might be associated with a higher likelihood of injury occurrence, indicating their potential as strong predictors for injury risk.

Moderate Correlated Features: Features like daily_load, fatigue, average_running_speed, soreness, and top_speed also show correlations with the injury column, though these correlations are slightly lower compared to the previously mentioned features. These moderate correlations indicate a notable association between these factors and injury risk, providing additional variables that may contribute to predicting injuries.

Weakly Correlated Features: Features such as stress, defensive_performance, Total_distance, readiness, injury_ts, team_performance, mood, sleep_quality, offensive_performance, HIR, and

sleep_duration have weak correlations with the injury column. These variables may have limited predictive power for injury occurrence based on their correlation coefficients with the target variable, suggesting that they are less critical in predicting injuries.

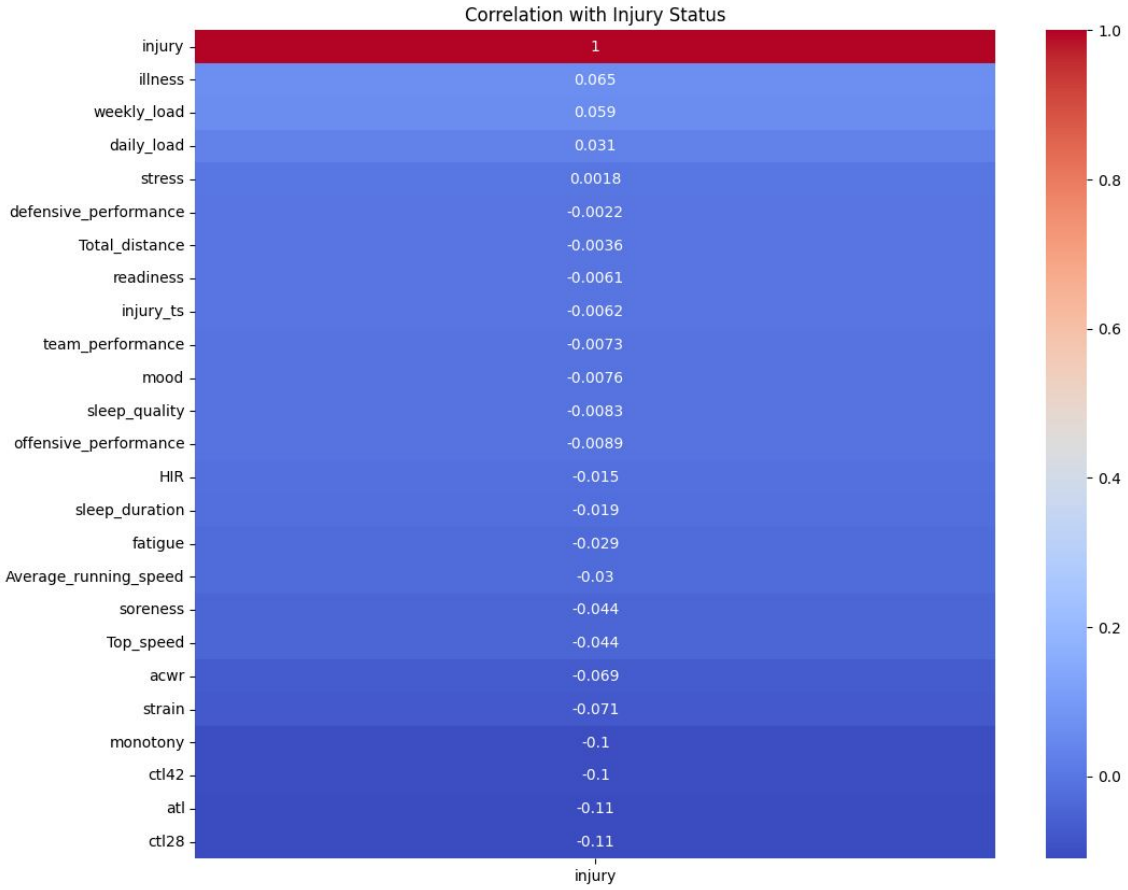

Figure 4.1: The figure presents information about the correlation between injury and other features.

Additionally, we observed several important correlations between features. There is a strong positive correlation (0.8999) between monotony and strain, indicating that increased monotony is associated with higher reported strain. Weekly load and daily load show a moderate positive correlation (0.5493), suggesting that athletes with higher weekly loads also tend to have higher daily loads. CTL42 and CTL28 are very strongly positively correlated (0.9416), highlighting their close relationship. Sleep quality and mood have a moderate positive correlation (0.4095), indicating that better sleep quality is linked to a better mood. Offensive and defensive performance are strongly positively correlated (0.6556), implying that they tend to improve or decline together. There is a moderate negative correlation (-0.1649) between weekly load and stress, suggesting that higher weekly loads might be associated with lower stress levels.

HIR and daily load show a moderate positive correlation (0.3404), indicating that higher daily loads may involve more high-intensity running. Stress and mood have a strong positive correlation (0.5958), showing that they tend to move in the same direction. Sleep quality and stress have a moderate positive correlation (0.4074), suggesting that higher stress levels may be linked to

poorer sleep quality. Readiness and sleep quality have a moderate positive correlation (0.1714), indicating that higher readiness might be associated with better sleep quality. Moderate negative correlations (-0.4073 and -0.3749) between ACWR and CTL42/CTL28 indicate an inverse relationship between the acute-to-chronic workload ratio and CTL42/CTL28. Lastly, injury timestamps and HIR have a weak negative correlation (-0.0205), implying a slight inverse relationship between injury timestamps and the Health Impact Ratio.

These findings underscore the complex interplay of various factors in injury risk, providing a comprehensive understanding of the key risk factors. The insights gained from this correlation analysis are crucial for developing more effective injury prevention strategies and tailoring training programs to mitigate injury risk.

4.2 Experiment 2: Optimal Window Size

In this section, we investigate the impact of varying the input window size on the performance of injury prediction algorithms. The input window size refers to the number of previous time steps considered when predicting an injury, and optimizing this parameter is crucial for capturing relevant temporal patterns in the data. We utilized a systematic approach to test different window sizes, ranging from 2 to 32, and evaluated their effects on algorithm performance using various oversampling and undersampling techniques to address the class imbalance.

The preprocessing function was designed to create sequences of player data, taking into account the specified window size. For each player in the dataset, sequences of data were generated, which were then aggregated to form the training and testing datasets. The function ‘create_team_sequence‘ ensured that the data for all players were included, allowing for comprehensive training of the algorithms.

To tackle the issue of class imbalance, we employed four different sampling methods: Random OverSampling, Random UnderSampling, SMOTE (Synthetic Minority Over-sampling Technique), and ADASYN (Adaptive Synthetic Sampling). Each method was implemented through dedicated functions: ‘oversample_data‘, ‘undersample_data‘, ‘smote_data‘, and ‘adasyn_oversample_data‘. These functions resampled the training data to balance the number of injuries and non-injury cases, aiming to improve the algorithms’ ability to detect injuries.

The experimental setup involved iterating over various combinations of hyperparameters:

- **Input Window Sizes**: 2, 4, 8, 16, 32
- **Test Sizes**: 0.2, 0.3
- **Oversampling Modes**: ”none”, ”oversample”, ”undersample”, ”smote”, ”adasyn”
- **Sampling Ratios**: 0.2, 0.3
- **Interpolate Injuries**: True, False

For each combination, the ‘preprocess‘ function was called to generate the corresponding training and testing datasets. The resampling was performed using the ‘sample_mode‘ function, which applied the chosen oversampling or undersampling technique based on the specified parameters.

The algorithms were then evaluated using the Naive Bayes classifier as a baseline. The evaluation metrics included accuracy, F1 score, precision, recall, and the confusion matrix. These metrics provided insights into how well each algorithm configuration handled the prediction task. The results for each configuration were recorded, including the percentage of zeros (non-injuries) in the dataset after resampling.

The code's structure ensured that each hyperparameter combination was systematically tested, allowing for a thorough comparison of the effects of different input window sizes and resampling techniques. This comprehensive approach facilitated the identification of the optimal window size and sampling strategy, leading to improved performance of the injury prediction algorithms.

The findings from these experiments are crucial for developing robust injury prediction algorithms that can effectively handle the temporal dynamics of the data and the inherent class imbalance. By fine-tuning the input window size and employing appropriate resampling methods, we aimed to enhance the algorithms' ability to predict injuries accurately and reliably, ultimately contributing to better injury prevention strategies in sports.

Models	Input Windows	Sampling Method	Recall	Precision	F1	Accuracy
Logistic Regression	16	SMOTE	0.25	0.28	0.26	0.99
Decision Tree	32	None	0.50	0.45	0.47	0.99
Random Forest	16	Undersample	0.72	0.09	0.16	0.95
K-Nearest Neighbor(KNN)	32	Oversample	0.60	0.24	0.34	0.98
LSTM	32	Undersample	0.10	1.0	0.18	0.99
Support Vector Machine (SVM)	8	None	0.27	0.27	0.27	0.99
XGBoost	32	SMOTE	0.50	0.71	0.58	0.99
Naive Bayes	16	SMOTE	0.75	0.06	0.12	0.93

Table 4.1: The optimal window size for injury prediction

4.3 Experiment 3: Top-tier Algorithms for Injury Prediction

In this section, we will conduct a thorough analysis of the performance of various machine learning algorithms in predicting injuries among soccer players. We will use the information from Table 4.1. to explain all eight algorithms performance in detail. Our evaluation will focus on key metrics such as the F1 score and confusion matrix to assess the accuracy and reliability of each algorithm. By scrutinizing these performance indicators, we aim to identify which algorithms excel in accurately predicting the occurrence of injuries. This analysis will provide valuable insights into the strengths and limitations of each algorithm, guiding us toward selecting the most effective approach for injury prediction in the context of sports analytics. Table A.6. provides a summary of the performance of these algorithms in correctly identifying injuries as injuries and injuries as non-injuries.

4.3.1 Logistic Regression

The performance of the Logistic Regression algorithm in predicting injuries is comprehensively summarized in Table 4.2. The algorithm successfully identified 4 actual injuries, correctly classifying these instances as injuries. However, it failed to recognize 12 actual injury cases, misclassifying them as non-injuries, which is a significant oversight. On the other hand, the algorithm incorrectly

predicted 10 non-injury cases as injuries, a type of error known as a false positive, while it accurately classified 2463 non-injury cases, correctly identifying them as non-injuries.

These results translated into a recall of 0.25, indicating that the algorithm correctly identified 25% of the actual injury cases. The precision of the algorithm was 0.28, meaning that only 28% of the cases it predicted as injuries were indeed injuries. The F1 score, which balances recall and precision, stood at 0.26, reflecting the algorithm's overall effectiveness in handling injury predictions. Despite these concerning metrics, the algorithm achieved a high overall accuracy of 0.99, primarily due to its ability to correctly classify the vast majority of non-injury cases.

While the high accuracy might suggest good performance, it largely reflects the algorithm's proficiency in predicting the majority class, which is non-injury cases. The low recall and precision scores highlight the algorithm's struggle to correctly identify injury cases, emphasizing the challenge of predicting rare events within a heavily imbalanced dataset. These metrics underscore the necessity of employing more sophisticated techniques or different algorithms to better capture the nuances of injury prediction. Addressing these issues is crucial for improving the algorithm's reliability and effectiveness in real-world applications where accurately predicting injuries can have significant implications.

	Predicted	
	Predicted Injuries	Predicted Non-Injuries
Actual Injuries	4	12
Actual Non-Injuries	10	2463

Table 4.2: Actual vs predicted injuries of Logistic Regression.

4.3.2 Decision Tree

The performance of the Decision Tree algorithm in predicting injuries is detailed in Table 4.3. The algorithm correctly identified 5 actual injury cases, classifying them as predicted injuries, while it misclassified an equal number of actual injuries, failing to recognize them and classifying them as non-injuries. Additionally, the algorithm incorrectly predicted 6 non-injury cases as injuries, but it accurately identified 1586 non-injury cases, classifying them correctly as non-injuries.

These results yielded a recall of 0.50, indicating that the algorithm successfully identified 50% of the actual injury cases. The precision of the algorithm was also 0.45, meaning that half of the cases it predicted as injuries were indeed injuries. The F1 score, which provides a balance between recall and precision, was 0.47, suggesting a moderate overall performance in injury prediction. The algorithm achieved a high overall accuracy of 0.99, reflecting its strong capability in correctly identifying non-injury cases.

While the accuracy is impressive, primarily driven by the large number of non-injury cases, the recall and precision scores suggest that the algorithm is equally split in its ability to identify true positives and false positives. This performance underscores the challenge of developing algorithms that can accurately predict rare events like injuries in a dataset dominated by non-injury cases. Despite these challenges, the Decision Tree algorithm's balanced recall and precision indicate a promising direction for further refinement and improvement in injury prediction.

	Predicted	
	Predicted Injuries	Predicted Non-Injuries
Actual Injuries	5	5
Actual Non-Injuries	6	1586

Table 4.3: Actual vs predicted injuries of Decision Tree.

4.3.3 Random Forest

The performance of the Random Forest algorithm in predicting injuries is illustrated in Table 4.4. The algorithm successfully identified 8 actual injury cases, accurately predicting them as injuries. However, it failed to recognize 3 injury cases, misclassifying them as non-injuries. The algorithm also incorrectly predicted 77 non-injury cases as injuries but correctly classified 1571 non-injury cases.

These results translated into a recall of 0.72, indicating that the algorithm correctly identified 72% of the actual injury cases, which is a notable improvement over some other algorithms. The precision was 0.09, reflecting the low proportion of true positive injury predictions among all predicted injury cases. The F1 score, balancing recall and precision, stood at 0.16, highlighting the algorithm's moderate performance in predicting injuries. The overall accuracy of the Random Forest algorithm was 0.95, demonstrating its high capability in accurately classifying non-injury cases.

Despite the relatively high recall, the low precision indicates a considerable number of false positives, where non-injury cases were incorrectly classified as injuries. This suggests that while the Random Forest algorithm is better at detecting injury cases compared to other algorithms, it still faces challenges in reducing the number of false positives. The algorithm's high accuracy primarily stems from its effectiveness in identifying non-injury cases, underscoring the complexities of injury prediction in an imbalanced dataset.

	Predicted	
	Predicted Injuries	Predicted Non-Injuries
Actual Injuries	8	3
Actual Non-Injuries	77	1571

Table 4.4: Actual vs predicted injuries of Random Forest.

4.3.4 K-Nearest Neighbors (KNN)

The performance of the K-Nearest Neighbors algorithm in predicting injuries is depicted in Table 4.5. The algorithm correctly identified 6 actual injury cases, classifying them accurately as injuries. However, it missed 4 injury cases, misclassifying them as non-injuries. On the flip side, the algorithm falsely predicted 19 non-injury cases as injuries while accurately classifying 1573 non-injury cases.

These outcomes translated into a recall of 0.60, indicating that the algorithm correctly identified 60% of the actual injury cases, which shows a relatively balanced performance. The precision was 0.24, reflecting that 24% of the cases predicted as injuries were true injuries. The F1 score, which

balances recall and precision, stood at 0.34, indicating moderate effectiveness in handling injury predictions. The overall accuracy of the KNN algorithm was 0.98, underscoring its high proficiency in classifying non-injury cases correctly.

While the KNN algorithm demonstrates a fair balance in identifying injury cases compared to some other algorithms, its precision indicates a moderate rate of false positives, where non-injury cases are misclassified as injuries. This algorithm's high accuracy is largely due to its strong performance in recognizing non-injury cases, highlighting the ongoing challenge of accurately predicting rare events such as injuries in a dataset with significant class imbalance.

	Predicted	
	Predicted Injuries	Predicted Non-Injuries
Actual Injuries	6	4
Actual Non-Injuries	19	1573

Table 4.5: Actual vs predicted injuries of K-Nearest Neighbour (KNN).

4.3.5 LSTM

The performance of the LSTM in predicting injuries is summarized in Table 4.1. The LSTM algorithm achieved a recall of 0.10, meaning it correctly identified only 10% of actual injury cases. Despite this, it had a precision of 1.0, indicating that all the cases it predicted as injuries were indeed injuries. This high precision suggests that while the algorithm was very cautious and selective in predicting injuries, it was highly accurate when it did make a prediction.

The F1 score for the LSTM algorithm was 0.18, reflecting a significant imbalance between recall and precision. The overall accuracy of the algorithm was 0.99, highlighting its effectiveness in correctly classifying the majority of non-injury cases.

These results show that while the LSTM algorithm is very precise in its injury predictions, its low recall indicates it misses a significant number of injury cases. This suggests that while the algorithm is conservative in its injury predictions to avoid false positives, it does so at the cost of failing to identify a substantial number of actual injuries, which is a crucial aspect in the context of injury prediction.

4.3.6 Support Vector Machine (SVM)

The performance of the Support Vector Machine algorithm in predicting injuries is illustrated in Table 4.6. The algorithm correctly identified 3 actual injury cases, classifying them accurately as injuries. However, it missed 8 injury cases, misclassifying them as non-injuries. On the other hand, the algorithm falsely predicted 8 non-injury cases as injuries while accurately classifying 1669 non-injury cases.

These outcomes translate into a recall of 0.27, indicating that the algorithm correctly identified 27% of the actual injury cases, which shows a relatively low performance in detecting injuries. The precision was 0.27, reflecting that 27% of the cases predicted as injuries were true injuries. The F1 score, which balances recall and precision, stood at 0.27, indicating limited effectiveness in

handling injury predictions. The overall accuracy of the SVM algorithm was 0.99, underscoring its high proficiency in classifying non-injury cases correctly.

While the SVM algorithm demonstrates high accuracy, primarily due to its strong performance in recognizing non-injury cases, its low recall and precision highlight the challenge of accurately predicting rare events such as injuries in a dataset with significant class imbalance. The algorithm's high accuracy is misleading when considering its performance on the minority class (injuries), emphasizing the need for strategies to improve the identification of such rare but critical events.

	Predicted	
	Predicted Injuries	Predicted Non-Injuries
Actual Injuries	3	8
Actual Non-Injuries	8	1669

Table 4.6: Actual vs predicted injuries of Support Vector Machine (SVM).

4.3.7 XGBoost

The performance of the XGBoost algorithm in predicting injuries is depicted in Table 4.7. The algorithm correctly identified 5 actual injury cases, classifying them accurately as injuries. However, it missed 5 injury cases, misclassifying them as non-injuries. On the other hand, the algorithm falsely predicted 2 non-injury cases as injuries while accurately classifying 1590 non-injury cases.

These outcomes translated into a recall of 0.50, indicating that the algorithm correctly identified 50% of the actual injury cases, showing a balanced performance in detecting injuries. The precision was 0.71, reflecting that 71% of the cases predicted as injuries were true injuries. The F1 score, which balances recall and precision, stood at 0.58, indicating fairly effective handling of injury predictions. The overall accuracy of the XGBoost algorithm was 0.99, underscoring its high proficiency in classifying non-injury cases correctly.

The XGBoost algorithm demonstrates a commendable balance in identifying injury cases compared to other algorithms, with a higher precision and recall. This algorithm's high accuracy is primarily due to its strong performance in recognizing non-injury cases, while its improved recall and precision highlight its effectiveness in handling the challenge of accurately predicting rare events such as injuries in a dataset with significant class imbalance.

	Predicted	
	Predicted Injuries	Predicted Non-Injuries
Actual Injuries	5	5
Actual Non-Injuries	2	1590

Table 4.7: Actual vs predicted injuries of XGBoost.

4.3.8 Naive Bayes

The performance of the Naive Bayes algorithm in predicting injuries is illustrated in Table 4.8. The algorithm correctly identified 12 actual injury cases, classifying them accurately as injuries.

However, it missed 4 injury cases, misclassifying them as non-injuries. Conversely, the algorithm falsely predicted 164 non-injury cases as injuries while accurately classifying 2309 non-injury cases.

These outcomes translated into a recall of 0.75, indicating that the algorithm correctly identified 75% of the actual injury cases, demonstrating a strong performance in detecting injuries. The precision was 0.06, reflecting that only 6% of the cases predicted as injuries were true injuries. The F1 score, which balances recall and precision, stood at 0.13, indicating limited effectiveness in handling injury predictions despite the high recall. The overall accuracy of the Naive Bayes algorithm was 0.93, showing good proficiency in classifying non-injury cases correctly.

While the Naive Bayes algorithm exhibits high recall, indicating its effectiveness in identifying most injury cases, its precision highlights a significant rate of false positives, where non-injury cases are misclassified as injuries. This algorithm's overall accuracy is relatively high due to its performance in recognizing non-injury cases. However, the low precision underscores the challenge of accurately predicting rare events such as injuries, revealing a need for improvement in reducing false positives.

	Predicted	
	Predicted Injuries	Predicted Non-Injuries
Actual Injuries	12	4
Actual Non-Injuries	164	2309

Table 4.8: Actual vs predicted injuries of Naive Bayes.

4.4 Experiment 4: Impact of Hyperparameter Tuning

In this section, we explore the effects of hyperparameter tuning on the performance of various machine learning algorithms used for injury prediction in soccer players. Hyperparameter tuning is a critical step in algorithm optimization that involves adjusting the parameters of an algorithm to improve its performance. By employing techniques such as grid search cross-validation and different sampling methods to address class imbalance, we aimed to enhance the predictive accuracy and reliability of our algorithms. Table 4.9 presents the results after hyperparameter tuning for Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbor, Support Vector Machine, XGBoost and Naive Bayes algorithms.

Logistic Regression: For the Logistic Regression algorithm, we fine-tuned parameters including the penalty type ('l1' or 'l2'), regularization strength 'C' (ranging from 0.001 to 1000), and solver ('liblinear' or 'saga'). The grid search cross-validation process identified the optimal combination of these parameters, resulting in a recall of 0.25, a precision of 0.33, and an F1 score of 0.28, with an overall accuracy of 0.99. The tuning improved precision compared to the untuned algorithm, indicating a better identification of true positives. However, the recall and F1 scores remained low, reflecting the ongoing challenge of detecting injury cases in a highly imbalanced dataset. The use of the Synthetic Minority Over-sampling Technique (SMOTE) helped to balance the dataset, but predicting rare events such as injuries continued to be a significant hurdle.

Decision Tree: The Decision Tree algorithm's hyperparameters included the criterion ('gini' or 'entropy'), maximum depth (None, 10, 20, 30), minimum samples split (2, 5, 10), minimum samples

leaf (1, 2, 4), and maximum features (‘auto‘, ‘sqrt‘, ‘log2‘). After applying grid search for optimal parameter selection, the Decision Tree algorithm achieved a recall and precision of 0.40 and 0.57, respectively, an F1 score of 0.47, and an accuracy of 0.99. This balanced performance demonstrates that hyperparameter tuning and the use of oversampling effectively addressed class imbalance, enabling the algorithm to identify both injury and non-injury cases with similar accuracy. The balanced metrics suggest a well-calibrated algorithm capable of managing the complexities of the injury prediction task.

Random Forest: We fine-tuned Random Forest algorithm setting parameters such as the number of trees or ’n_estimators’ ranging from 100 to 300, the maximum tree depth as None, 10, 20, and 30, the minimum samples required to split a node or ’min_samples_split’ of 2, 5, 10, the minimum samples required at a leaf node or ’min_samples_leaf’, of 1, 2, 4, and the number of features considered for splitting or ’max_features’ set to ’auto’, ’sqrt’, ’log2’. We have identified the optimal parameter combination using the Randomized Search Cross-Validation. Then the algorithm performed 0.20 F1 score with 0.72 recall and 0.11 precision showing high recall with low precision indicating a high rate of positives. It shows the difficulty in achieving balanced performance for injury prediction.

K-Nearest Neighbor (KNN): The K-Nearest Neighbors algorithm was tuned for parameters such as the number of neighbors (3, 5, 7, 9, 11), weight function (‘uniform‘ or ‘distance‘), and the parameter ‘p‘ for the Minkowski distance metric (1 or 2). The optimal settings resulted in a recall of 0.30, a precision of 1.0, an F1 score of 0.46, and an accuracy of 0.99. The exceptionally high precision indicates that the KNN algorithm, when predicting injuries, is almost always correct, thus minimizing false positives. However, the relatively lower recall suggests that while the algorithm is accurate, it is conservative in predicting injuries, leading to some missed injury cases. This trade-off between precision and recall is crucial in contexts where the cost of false positives or negatives can have significant implications.

Support Vector Machine (SVM): We have fine-tuned the SVM algorithm also setting several parameters. We have used ’C’ with the values of 0.1, 1, 10, 100, kernel type as ’linear’, ’poly’, ’rbf’, ’sigmoid’, and kernel coefficient ’gamma’ as ’scale’ or ’auto. After applying Randomized Search Cross-Validation, the best parameters yielded a recall of 0.18, precision of 0.60, F1 score of 0.28, and accuracy of 0.99. The performance showcases it has avoided false positives but the lower recall proofs it has missed many true injury cases.

XGBoost: We fine tuned XGBoost algorithm with several parameters. The parameters we used are ’scale_pos_weight’, learning rate (0.01, 0.05, 0.1, 0.2), maximum depth(’max_depth’ of 3, 5, 7, 10), minimum child weight (’min_child_weight’ of 1, 3, 5), ’gamma’ (0, 0.1, 0.2, 0.3), subsample ratio ’subsample’ (0.6, 0.8, 1.0), ’colsample_bytree’ (0.6, 0.8, 1.0), ’reg_alpha’ and ’reg_lambda’ (0, 0.1, 1.0). This optimization performed 0.57 F1 score with 0.40 recall and 1.0 precision.

Naive Bayes: The Naive Bayes algorithm was optimized by tuning the smoothing parameter ‘alpha‘ (values of 0.1, 0.5, 1.0, 2.0). Post-tuning, the algorithm demonstrated a recall of 0.60, a precision of 0.09, an F1 score of 0.16, and an accuracy of 0.96. The high recall suggests that the algorithm is effective at identifying injury cases, but the low precision and F1 score indicate a high number of false positives. This reflects an algorithm that is highly sensitive to detecting injuries but lacks specificity, resulting in many incorrect injury predictions. This performance underscores

the need for a careful balance between sensitivity (recall) and precision to ensure meaningful and actionable predictions.

Overall, the results highlight the significance of hyperparameter tuning and appropriate sampling methods in enhancing the performance of machine learning algorithms for injury prediction. Each algorithm responded differently to these adjustments, demonstrating the necessity for tailored approaches to optimize their predictive capabilities. Hyperparameter tuning proved essential in refining algorithm parameters and improving their ability to handle the imbalanced nature of injury data, ultimately leading to more accurate and reliable injury prediction algorithms. This comprehensive approach ensures that the algorithms are well-equipped to manage the complexities and nuances of predicting rare events such as injuries in soccer players.

Models	Input Windows	Sampling Method	Recall	Precision	F1	Accuracy
Logistic Regression	16	SMOTE	0.25	0.33	0.28	0.99
Decision Tree	32	Oversample	0.40	0.57	0.47	0.99
Random Forest	16	Undersample	0.72	0.11	0.20	0.96
K-Nearest Neighbor(KNN)	32	None	0.30	1.0	0.46	0.99
Support Vector Machine (SVM)	16	None	0.18	0.60	0.28	0.99
XGBoost	32	SMOTE	0.40	1.0	0.57	0.99
Naive Bayes	32	None	0.60	0.09	0.16	0.96

Table 4.9: Algorithms performance after hyper tuning

4.5 Experiment 5: Team Specific Injury Forecasting

In this section, we focus on the performance of injury prediction algorithms when applied specifically to a single team, namely Team A. The goal is to determine whether tailoring algorithms to a particular team can enhance prediction accuracy compared to using a general algorithm for multiple teams. The results, summarized in Table 4.10, indicate that several algorithms exhibit improved performance when calibrated specifically for Team A.

The algorithms were evaluated using the same hyperparameter tuning and resampling techniques as in previous experiments. The input window size was set to 32 for most algorithms, with the Naive Bayes algorithm utilizing an 8-window configuration. Various resampling methods such as Adasyn, undersampling, oversampling, and none were employed to address class imbalance.

Key observations from the team-specific results for Team A include:

Logistic Regression: with Adasyn achieved a recall of 0.20 and a precision of 0.25, leading to an F1 score of 0.22 and an accuracy of 0.99. While the accuracy is high, the recall indicates a need for better identification of injury cases, suggesting that the algorithm can be refined further to balance the trade-off between precision and recall.

Decision Tree: with no resampling exhibited improved performance with a recall of 0.30 and a precision of 0.42, resulting in an F1 score of 0.35 and an accuracy of 0.99. This indicates that the algorithm can effectively balance between detecting injuries and avoiding false positives.

Random Forest: using undersampling, showed a significantly high recall of 0.80 but a low precision of 0.14, leading to an F1 score of 0.25 and an accuracy of 0.96. This algorithm's high recall suggests it is particularly good at identifying most injury cases, albeit with many false positives. This is useful in scenarios where it is crucial to identify as many injuries as possible, even at the cost of some false alarms.

K-Nearest Neighbor (KNN): with no resampling, achieved a balanced performance with a recall of 0.40 and a high precision of 0.66, resulting in an F1 score of 0.50 and an accuracy of 0.99. This indicates a good balance between identifying true injury cases and minimizing false positives, making it a strong candidate for practical application in injury prediction.

LSTM: with oversampling reached a high recall of 0.80 but a low precision of 0.08, which led to an F1 score of 0.14 and an accuracy of 0.93. This suggests the algorithm is good at detecting injuries but suffers from a high rate of false positives, indicating a need for further tuning to improve precision.

Support Vector Machine (SVM): without resampling, had a recall of 0.20 and a precision of 0.33, culminating in an F1 score of 0.25 and an accuracy of 0.99. This algorithm shows moderate performance, with room for improvement in both recall and precision.

XGBoost: with Adasyn delivered a recall of 0.30 and an impressive precision of 0.60, resulting in an F1 score of 0.40 and the highest accuracy of 0.99 among the algorithms tested. This algorithm's strong performance highlights its potential as a reliable tool for injury prediction when optimized for specific team data.

Naive Bayes: using Adasyn and an input window of 8, achieved a recall of 0.54 and a precision of 0.13, leading to an F1 score of 0.21 and an accuracy of 0.96. While the recall is relatively high, the low precision indicates a high rate of false positives, suggesting that this algorithm may benefit from further refinement and possibly different resampling strategies.

These results demonstrate that algorithm performance can be significantly improved when algorithms are tailored to specific teams. For instance, the KNN algorithm showed a notably higher F1 score when applied to Team A compared to its general performance across multiple teams. Similarly, the Random Forest algorithm, despite its lower precision, provided an excellent recall rate, making it a valuable tool for injury detection in scenarios where missing an injury is more critical than having false positives.

Overall, these findings suggest that team-specific algorithms can offer enhanced predictive capabilities, allowing for more precise and reliable injury forecasts. Such tailored approaches are essential for optimizing player health management and preventing injuries, thereby improving team performance and player well-being. Additionally, this experiment highlights the importance of considering specific team dynamics and characteristics in the development and application of predictive algorithms. By focusing on the unique attributes and data of each team, it is possible to achieve a higher level of accuracy and effectiveness in injury prediction, ultimately leading to better outcomes in sports performance and athlete health management.

Models	Input Windows	Sampling Method	Recall	Precision	F1	Accuracy
Logistic Regression	32	Adasyn	0.20	0.25	0.22	0.99
Decision Tree	32	None	0.30	0.42	0.35	0.99
Random Forest	32	Undersample	0.80	0.14	0.25	0.96
K-Nearest Neighbor(KNN)	32	None	0.40	0.66	0.50	0.99
LSTM	32	Oversample	0.80	0.08	0.14	0.93
Support Vector Machine (SVM)	32	None	0.20	0.33	0.25	0.99
XGBoost	32	Adasyn	0.30	0.60	0.40	0.99
Naive Bayes	8	Adasyn	0.54	0.13	0.21	0.96

Table 4.10: Results for Specific Team (Team A).

4.6 Experiment 6: Injured Players Injury Forecasting

In this experiment, we focused on enhancing the predictive capability of our algorithms by exclusively utilizing data from players with a history of injuries. By removing data from players who have never been injured, the algorithms were provided with a clearer and more consistent pattern of injury occurrences, allowing them to better learn and predict injuries.

Table 4.11 summarizes the performance metrics of various algorithms when applied to this injury-specific dataset. The following observations highlight the efficacy of each algorithm:

Logistic Regression: with a 16-window input and SMOTE for oversampling achieved a recall of 0.25, precision of 0.28, F1 score of 0.26, and an accuracy of 0.99. This indicates a moderate balance between sensitivity and precision, suggesting that while the algorithm is fairly accurate in general, it could benefit from further tuning to improve its ability to correctly identify injuries.

Decision Tree: using a 32-window input and Adasyn, significantly improved its recall to 0.70, with a precision of 0.36, resulting in an F1 score of 0.48 and an accuracy of 0.99. This indicates that the Decision Tree algorithm is adept at identifying injury instances with a reasonable trade-off between false positives and false negatives, making it a reliable choice for injury prediction.

Random Forest: with a 16-window input and undersampling showed a high recall of 0.81 but a low precision of 0.08, leading to an F1 score of 0.15 and an accuracy of 0.94. The high recall suggests the algorithm is effective at identifying injuries but suffers from a high rate of false positives, reducing its overall precision. This algorithm's tendency to predict more false positives highlights the need for further refinement to balance sensitivity and specificity.

K-Nearest Neighbor (KNN): utilizing a 32-window input and oversampling, achieved a recall of 0.60 and precision of 0.24, resulting in an F1 score of 0.34 and an accuracy of 0.98. This algorithm strikes a balance, being moderately effective at predicting injuries with a lower false positive rate compared to Random Forest. However, there is room for improvement in terms of precision.

LSTM: also with a 32-window input and oversampling, reached a recall of 0.66 and a precision of 0.06, leading to an F1 score of 0.10 and an accuracy of 0.93. The high recall indicates that the algorithm is good at detecting injuries, but the low precision suggests a high number of false positives. Further refinement of the LSTM algorithm is needed to enhance its precision and overall performance.

Support Vector Machine (SVM): with an 8-window input and no resampling, achieved equal recall and precision of 0.27, resulting in an F1 score of 0.27 and an accuracy of 0.99. This balanced performance suggests that the algorithm is consistent but could benefit from additional tuning or different resampling methods to enhance its predictive power.

XGBoost: with a 32-window input and SMOTE displayed strong performance with a recall of 0.50 and a high precision of 0.71, resulting in an F1 score of 0.58 and an accuracy of 0.99. This high precision and accuracy make XGBoost particularly effective for this dataset, indicating its robustness in predicting injuries with fewer false positives. XGBoost's performance highlights its potential as a primary tool for injury forecasting.

Naive Bayes: using a 16-window input and SMOTE, achieved a high recall of 0.75 but a low precision of 0.06, leading to an F1 score of 0.12 and an accuracy of 0.93. This algorithm effectively identifies most injuries but at the cost of a high false positive rate, suggesting that further tuning is necessary to improve precision. The high recall is promising, but the algorithm's performance could be significantly enhanced with better precision.

Overall, by focusing on data from injury-prone players, the algorithms were able to better identify injury patterns, resulting in varying levels of success across different algorithms. The XGBoost algorithm stands out with its high precision and accuracy, making it a promising tool for injury prediction in scenarios where minimizing false positives is crucial. Meanwhile, the Decision Tree and Random Forest algorithms also showed substantial recall, indicating their potential usefulness in contexts where capturing as many injury cases as possible is a priority.

This experiment underscores the importance of tailored data preprocessing and algorithm selection to enhance injury prediction accuracy. By concentrating on injury-prone players, we can develop more specialized and effective predictive algorithms, ultimately contributing to better player management and injury prevention strategies. Such refined algorithms can be pivotal in the sports industry, helping teams to preemptively manage player health, optimize performance, and potentially extend the careers of athletes.

Models	Input Windows	Sampling Method	Recall	Precision	F1	Accuracy
Logistic Regression	16	SMOTE	0.25	0.28	0.26	0.99
Decision Tree	32	Adasyn	0.70	0.36	0.48	0.99
Random Forest	16	Undersample	0.81	0.08	0.15	0.94
K-Nearest Neighbor(KNN)	32	Oversample	0.60	0.24	0.34	0.98
LSTM	32	Oversample	0.66	0.06	0.10	0.93
Support Vector Machine (SVM)	8	none	0.27	0.27	0.27	0.99
XGBoost	32	SMOTE	0.50	0.71	0.58	0.99
Naive Bayes	16	SMOTE	0.75	0.06	0.12	0.93

Table 4.11: Performance of the models on the only injured players dataset.

4.7 Experiment 7: Dataset Scale

In this section, we examine how the scale of the dataset influences the performance of injury prediction algorithms. By exploring different scales, including the full dataset, data specific to a

single team (Team A), and data solely from players with a history of injuries, we aim to understand the impact of dataset size and composition on algorithm efficacy.

4.7.1 Full Dataset

The full dataset comprises all available data points, providing a broad view of injury and non-injury instances across all teams. The distribution of injury cases in the full dataset is as follows:

- Non-injury cases: 8526
- Injury cases: 56'

Using the full dataset allows the algorithms to learn from a comprehensive set of examples, potentially capturing a wide range of injury patterns. However, the high imbalance between non-injury and injury cases poses a significant challenge for the algorithms to accurately predict injuries. This imbalance can lead to an algorithm that is biased toward predicting non-injury cases, thereby reducing its effectiveness in identifying actual injury risks.

4.7.2 Single Team Dataset

For the single team dataset, we focus exclusively on Team A. This subset is smaller but more homogeneous, providing insights into the team's specific injury patterns. The distribution of injury cases for Team A is:

- Non-injury cases: 7180
- Injury cases: 56

By isolating Team A's data, we aim to tailor the algorithms more closely to the unique characteristics and injury trends of this specific team. This approach allows for more precise injury predictions within the team context but might limit the generalizability of the algorithms to other teams. The homogeneity of the data can enhance the algorithm's sensitivity to injury patterns specific to Team A, but it may also result in overfitting to the nuances of this particular team.

4.7.3 Only Injured Players Dataset

This subset includes only the data from players who have previously experienced injuries, offering a focused view on those with a higher likelihood of recurring injuries. The distribution of injury cases in this dataset is:

- Non-injury cases: 3134
- Injury cases: 56

Concentrating on players with a history of injuries provides a clearer and more consistent pattern for the algorithms to learn from, potentially improving their ability to predict future injuries in these high-risk individuals. However, this approach excludes data from non-injured players, which might limit the overall predictive power regarding new injury cases. The focused nature of this dataset helps in understanding the recurring injury trends, but it might miss out on broader patterns that include the transition from non-injury to injury status.

The scale and composition of the dataset significantly impact the performance of injury prediction algorithms. Each dataset variation offers unique advantages and challenges:

Full Dataset: Provides a comprehensive learning base but suffers from severe class imbalance, making it challenging to predict injuries accurately.

Single Team Dataset: Allows for more team-specific predictions but may lack generalizability, potentially limiting its application to other teams.

Only Injured Players Dataset: Enhances learning from high-risk individuals but may miss broader injury patterns that include players transitioning from non-injury to injury status.

By analyzing these different scales, we can determine the optimal dataset configuration for developing robust and accurate injury prediction algorithms. This experimentation helps identify the best strategies to manage class imbalance and tailor algorithms to specific contexts, ultimately leading to improved injury prevention and management in sports. This comprehensive approach is crucial for developing predictive algorithms that not only identify potential injuries but also offer actionable insights for player health management and injury prevention strategies, thus enhancing overall team performance and player well-being.

4.8 Chapter Summary

In this chapter, we explored various aspects of injury prediction in sports, focusing specifically on soccer players. We began by presenting some important features such as illness, weekly_load, ctl28, atl, ctl42, monotony, strain, and acwr that are highly correlated with injury. We have also highlighted moderately positive and weakly correlated features with injury. Next, we examine the impact of different input window sizes on algorithm performance, systematically testing a range of sizes and employing various resampling techniques to address class imbalance. This approach helped improve the algorithms' ability to capture relevant temporal patterns and handle the inherent skew in injury versus non-injury cases.

Moreover, this chapter represents information about the eight machine learning algorithms performance that we have used to forecast the injured Norwegian women's soccer players injuries. Furthermore, we delved into hyperparameter tuning and its effect on algorithm performance. By fine-tuning parameters for Logistic Regression, Decision Tree, K-Nearest Neighbor, and Naive Bayes algorithms using grid search cross-validation, we enhanced their predictive accuracy and reliability. The implementation of resampling methods like Random OverSampling, Random UnderSampling, SMOTE, and ADASYN further balanced the datasets, contributing to better algorithm performance.

We also investigated the benefits of tailoring algorithms to specific contexts, such as team-specific injury prediction. Focusing on Team A, we found that algorithms calibrated for this team showed improved accuracy and precision, highlighting the advantage of using homogeneous and context-specific data. This approach underscored the potential for more precise injury forecasts when algorithms are customized to the unique characteristics and injury patterns of a specific team.

Furthermore, we assessed the impact of using data solely from players with a history of injuries.

This injury-focused dataset allowed the algorithms to learn from clearer and more consistent injury patterns, improving their ability to predict future injuries in high-risk individuals. However, this approach also illuminated the challenge of potentially missing broader injury patterns that might be present in a more diverse dataset.

Lastly, we compared different dataset scales to understand how the size and composition of the data influence algorithm performance. We examined the full dataset, a single team's dataset, and data exclusively from injured players. Each scale offered unique insights and challenges: the full dataset provided a comprehensive learning base but suffered from severe class imbalance; the single team's dataset allowed for more precise, context-specific predictions but might lack generalizability; and the dataset of only injured players enhanced learning from high-risk individuals but excluded potentially valuable data from non-injured players.

Collectively, these experiments provided valuable insights into optimizing injury prediction algorithms. By understanding the effects of various parameters and dataset configurations, we can develop more robust and accurate algorithms. This research contributes to better injury prevention and management strategies in sports, ultimately aiding in player health, performance optimization, and injury mitigation.

The next chapter focused on our main research objective and sub-research question, and we have discussed our findings from the Chapter 4. We also highlighted insights from our experiments and lessons learned during our research study. Additionally, we briefly covered our limitations, future work, and our contributions. Furthermore, we have discussed the ethical considerations related to the data of Norwegian women's soccer players.

Chapter 5

Discussion

In this chapter, we explore the insights and lessons gleaned from our experiments. We will evaluate the performance of various machine learning algorithms in predicting injuries and discuss the broader implications of our findings. We will revisit and address the research questions posed in Section 1.2. providing comprehensive answers based on the results obtained. This includes exploring the key factors correlated with injuries, investigating the optimal time frame to predict the injury, examining the effectiveness of various machine learning algorithms, and exploring the impact of hyperparameter tuning. Additionally, we will explore the significance of tailoring algorithms to specific teams, the performance of algorithms on injured-player datasets, and the influence of dataset scale.

We will also delve into the practical applications and real-world use cases of our findings, showing how these insights can be utilized effectively. Additionally, we will address the limitations we faced during our research, suggest potential areas for future work, and discuss the ethical considerations involved in deploying machine learning algorithms for predicting sports injuries. Finally, a summary will recapitulate the key contributions of this study, emphasizing the advancements made and their relevance to the field of sports analytics.

5.1 Addressing the Research Questions

In this section, we revisit the research question presented in Section 1.2. and provide thorough answers based on our findings. Section 1.2. outlined one primary research objective and seven sub-research questions that were derived from it. By systematically addressing each sub-question, we build a comprehensive understanding of our main research topic. We then integrate these responses to present a cohesive and detailed answer to our primary research question, ensuring that all aspects of the research are covered and connected to provide a clear and insightful conclusion.

RQ1: What are the most important features that are correlated with injuries in women's soccer players?

Our study identified several key factors that correlate highly with injuries in women's soccer players, offering valuable insights for injury prevention strategies and player management. We have observed that features such as illness, weekly load, CTL28, ATL, CTL42, monotony, strain, and ACWR are highly correlated with injury. Figure 4.1. is the visual representation where we have identified

all of the highly correlated features with injury.

Furthermore, we also found strong correlations between monotony and strain, weekly load and daily load, and between 42-day and 28-day Chronic Training Load (CTL), emphasizing the importance of managing training consistency and volume. The positive relation between stress and mood highlights the need to address stress for both mental health and injury prevention. Additionally, the connection between readiness and sleep quality underscores the role of good sleep in maintaining athlete readiness and reducing injury risk. We also observed that maintaining balanced workload ratios, as indicated by moderate negative correlations between the Acute:Chronic Workload Ratio (ACWR) and CTL42/CTL28, is crucial to preventing injuries caused by sudden increases in training intensity. These findings underscore the complexity of injury risk in women's soccer and suggest that targeted interventions and monitoring can improve injury prevention strategies. Future research should delve into how these factors interact to enhance evidence-based approaches for injury prevention and player health management.

RQ2: How many days is the most effective time frame for predicting injuries?

In our thesis, we have used eight machine learning algorithms to predict injuries in Norwegian women's players. One of our main goals was to determine the most effective time frame for making these predictions. We have examined different window sizes, including 2, 4, 8, 16, and 32 days. From the Table C.1. we can see that the short-term windows (8–16) showed limited effectiveness. Algorithms several algorithms, Support Vector Machine performed best within an 8-day time frame where the F1 score is 0.27. On the other hand, Logistic Regression, Random Forest, and Naive Bayes performed best with an F1 score of 0.26, 0.16, and 0.12. Using a 16-day time frame, Random Forest and Naive Bayes performed poorly. If we looked into the 32-day time frame, four algorithms out of eight performed best within this time frame. XGBoost performed best with a 32-day time frame where the F1 score is 0.58 and recall and precision are 0.50 and 0.71, respectively. The decision tree showed a balanced performance with an F1 score of 0.47, whereas the recall and precision were 0.50 and 0.45, respectively. Figure A.5. highlights that 32-day window size is best to predict injury compared to other window sizes.

RQ3: Which machine learning algorithm is most effective for predicting injuries for the following day?

Our study investigated the effectiveness of various machine learning algorithms in predicting injuries. We have considered the F1 score to measure our algorithm's performance because it takes into account both precision and recall. We explored eight machine learning algorithms such as Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, LSTM, Support Vector Machine, XGBoost, and Naive Bayes.

With a 16-day input window with SMOTE sampling, we ran the Logistic Regression algorithm. The result was an F1 score of 0.26, which indicates decent skill but a significant amount of false positives. The Decision Tree algorithm, using a 32-day input window without sampling, performed better with an F1 score of 0.47, balancing recall and precision effectively.

We found that the Random Forest algorithm, using a 16-day input window with undersampling, had a high recall (0.72) but low precision (0.09), resulting in a poor F1 score of 0.16. Our K-Nearest

Neighbors algorithm, with a 32-day input window and oversampling, showed a better balance but still only reached an F1 score of 0.34.

Although the LSTM algorithm had perfect precision, its low recall (0.10) resulted in an F1 score of 0.18 due to its conservative predictions. On the other hand, Support Vector Machine have a balance recall, precision, and F1 score of 0.27 when the input window size is 8 with a sampling method of none.

XGBoost emerged as our top performer, using a 32-day input window with SMOTE sampling, achieving an F1 score of 0.58 with balanced recall (0.50) and high precision (0.71). Our Naive Bayes algorithm, with a 16-day input window and SMOTE sampling, had the lowest F1 score of 0.12, despite high recall (0.75), due to very low precision (0.06).

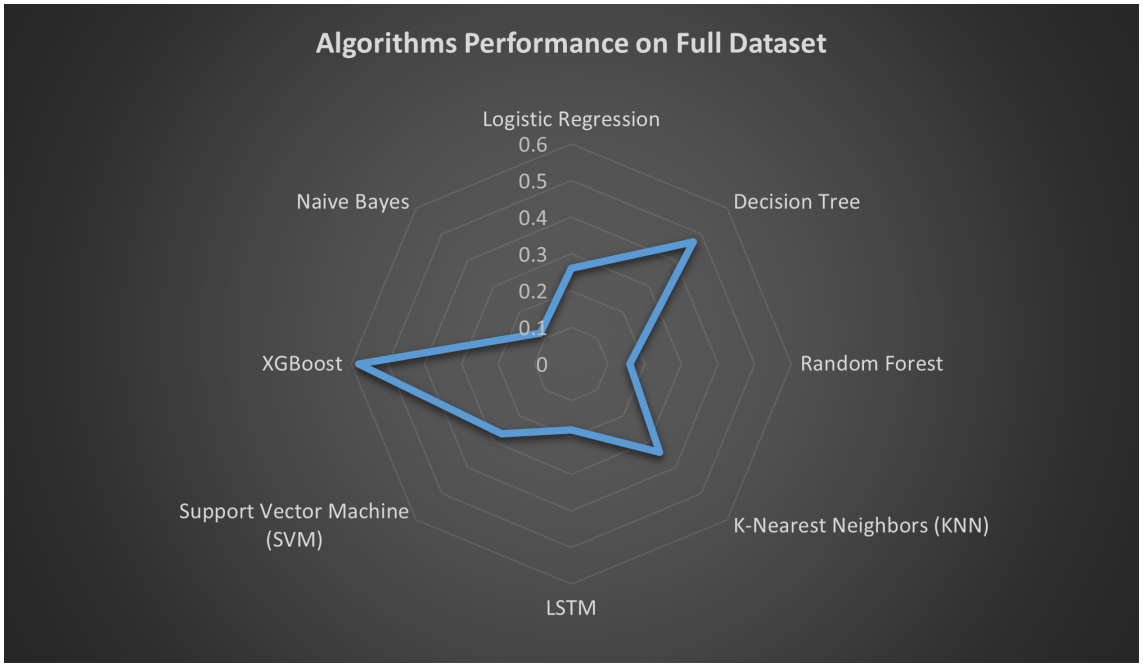

Figure 5.1: The radar chart helps in visualizing and comparing the relative performance of different machine learning models, allowing for easy identification of which models perform better or worse.

We found that the XGBoost algorithm was the best performer in predicting injuries with an F1 score of 0.58. However, the Decision Tree algorithm performed well with an F1 score of 0.47. While Random Forest and Naive Bayes had high recall, their low precision led to poor F1 scores. K-Nearest Neighbors and LSTM showed potential but require further tuning. We believe that future improvements could involve refining these algorithms or exploring hybrid approaches to enhance predictive accuracy and reliability.

RQ4: How does the hyperparameter tuning have an influence on improving the performance of the algorithms?

We applied hyperparameter tuning to improve the performance of machine learning algorithms. Figure 5.2. displayed the performance of the algorithms before and after hyperparameter tuning.

We have tuned the parameters of Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, Support Vector Machine, XGBoost, and Naive Bayes. After tuning, the logistic regression achieved an F1 score of 0.28 with a 16-day input window and SMOTE sampling, which is considered better than its previous performance. The Decision Tree algorithm, using Oversample sampling and a 32-day input window, scored 0.47, showing balanced recall and precision. However, the decision tree's performance did not improve after hyperparameter tuning, indicating that it did not achieve better results compared to its previous performance. The possible reasons are insufficient hyperparameter search, inherent limitations of the algorithm, a lack of informative features, and so on. The K-Nearest Neighbors, with a 32-day input window and no sampling, scored 0.46 but had high precision and lower recall.

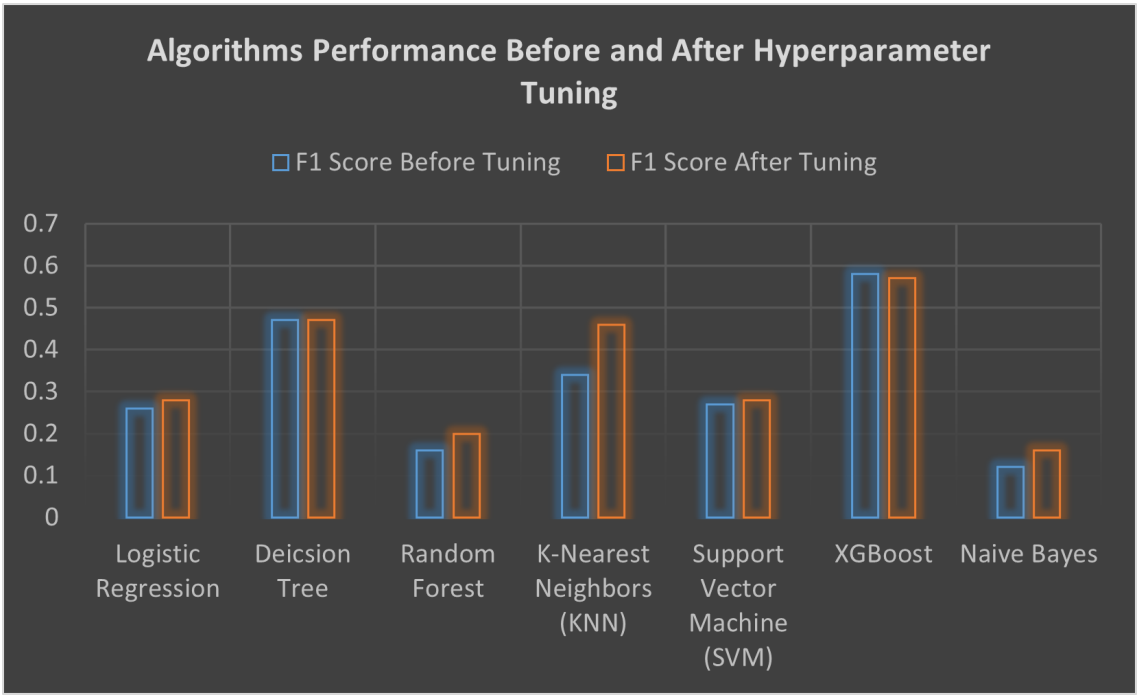

Figure 5.2: The graphs provide information about the machine learning model's performance before and after hyperparameter tuning.

The Naive Bayes algorithm, despite tuning, had a poor F1 score of 0.16 due to low precision. Random Forest, tuned with a 16-day input window and undersampling, scored 0.20 with high recall but low precision. Support Vector Machine, with a 16-day input window and no sampling, reached an F1 score of 0.28 by low recall and high precision.

XGBoost, our top performer, achieved an F1 score of 0.57 with a 32-day input window and SMOTE sampling, showing balanced and robust performance. Although the performance of XGBoost did not improve after tuning, indicating several possible reasons such as insufficient parameter search space, overfitting, a lack of features, and so on, this algorithm was the best at predicting injuries.

In summary, adjusting hyperparameters improved our algorithms performance, including Logistic Regression, Random Forest, Support Vector Machine, and Naive Bayes. The XGBoost and Decision Tree have a higher F1 score compared to other algorithms, although they require further tuning.

RQ5: Is it more effective to use a single algorithm for all teams or to develop separate algorithms for each team when predicting injuries?

Using a single algorithm for all teams is simpler and more scalable. It can learn from a diverse dataset and generalize well across different teams. Figure 5.3. refers to the performance of the algorithms on a Single Team (Team A) dataset. For example, our XGBoost algorithm performed well across all teams, achieving a high F1 score of 0.58. However, this approach may struggle with team-specific nuances, leading to more false positives, as seen with algorithms like Random Forest and Naive Bayes.

On the other hand, developing separate algorithms for each team allows us to tailor predictions to specific team characteristics, improving accuracy. For instance, our K-Nearest Neighbors algorithm tuned for Team A achieved an F1 score of 0.50, showing good balance. Yet, this approach requires more resources and may be challenging for teams with limited data.

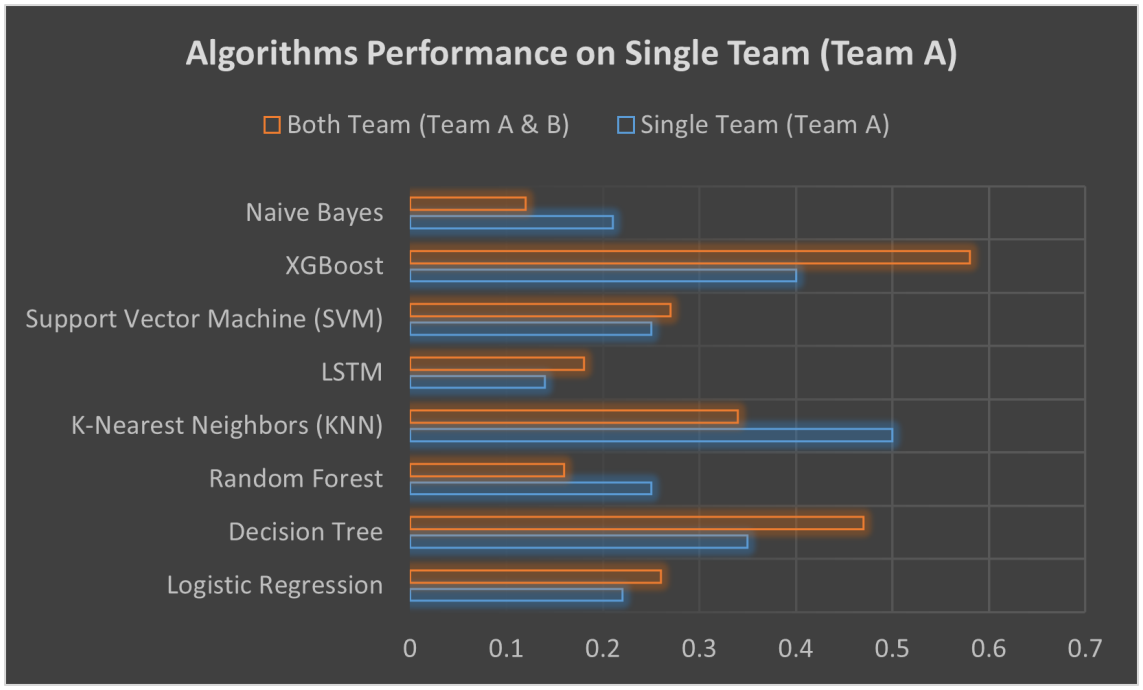

Figure 5.3: Machine learning models performance on a single team (Team A) dataset.

In conclusion, the choice depends on balancing generalization and accuracy. A single algorithm is practical for broad applications but may lack precision, while separate algorithms offer tailored insights but require more resources. Hybrid approaches could be explored in the future to combine the strengths of both methods.

RQ6: How do machine learning algorithms perform when evaluated on a dataset containing only injured players?

We looked at how well different machine learning algorithms can predict injuries in players who have already been injured. By focusing on this group, we aimed to understand how these

algorithms can identify patterns and predict future injuries in those at high risk. Figure 5.4. displayed the performance of machine learning algorithms on only injured players.

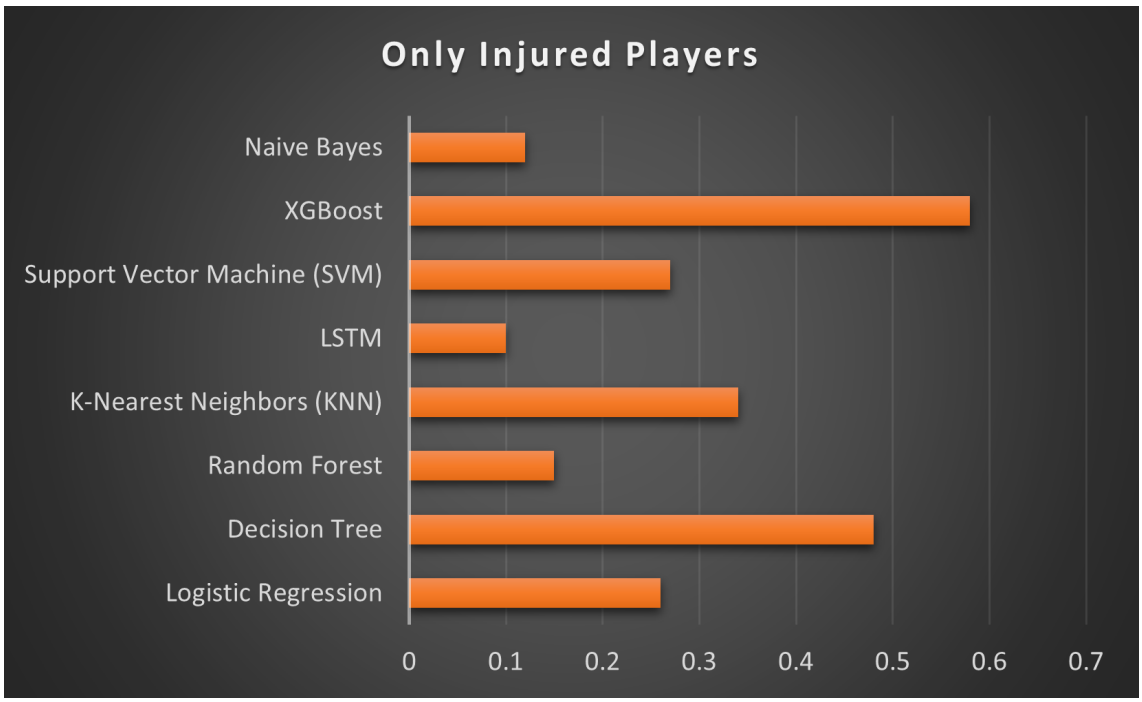

Figure 5.4: The bar chart gives information on the performance on the performance of seven machine learning models on the only injured players dataset.

Here's what we found: The Logistic Regression algorithm, using a 16-day input window with SMOTE sampling, got an F1 score of 0.26, showing it needs more fine-tuning to improve accuracy. The Decision Tree algorithm did better with a 32-day input window and Adasyn sampling, achieving an F1 score of 0.48, balancing recall and precision effectively. The Random Forest algorithm achieved a 0.15 F1 score with a 16-day input window and undersampling. K-Nearest Neighbors achieved 0.34 with a 32-day input window and oversampling. With a 32-day input window, the LSTM algorithm performed a 0.10 F1 score and oversampling due to many false positives. Using 8-day input Support Vector Machine scored 0.27 F1 score without sampling. XGBoost was the top performer with an F1 score of 0.58, using a 32-day input window with SMOTE sampling, offering the best balance between recall and precision. Naive Bayes, with a 16-day input window and SMOTE sampling, had the lowest F1 score of 0.12 due to very low precision. In summary, XGBoost and Decision Tree algorithms were the best at predicting injuries, highlighting the importance of algorithm selection and customization to improve prediction accuracy and player health management.

RQ7: How does the scale of the dataset have an impact on the performance of machine learning algorithms?

We examined how dataset size affects the performance of machine learning algorithms for predicting injuries. By analyzing three types of datasets—full, single team (Team A), and injured players only—we gained insights into their impact on prediction accuracy. Figure 5.5. showcases the performance on different datasets.

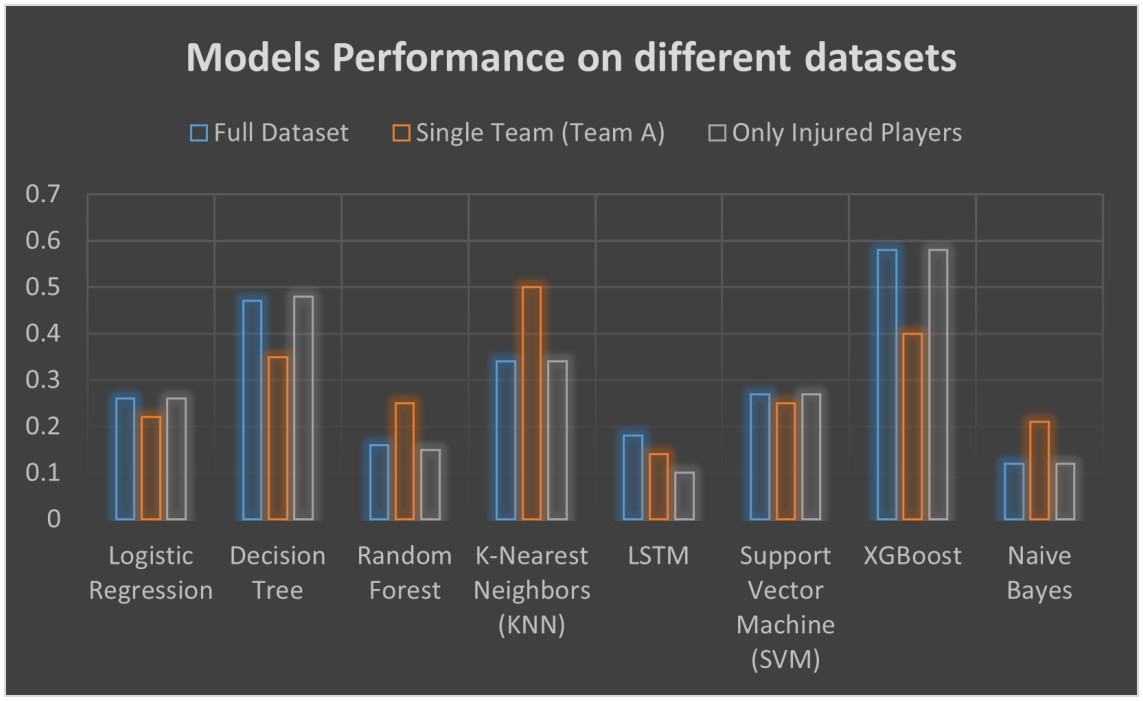

Figure 5.5: The line graphs illustrate information about the machine learning model's performance based on the different datasets.

The full dataset gave a broad view but struggled with class imbalance, making it harder to predict injuries accurately. Focusing on Team A's data offered more precise predictions within their context but might not generalize well to other teams. Concentrating on injured players provided clearer patterns but might miss out on broader injury trends. Each dataset had its pros and cons, influencing the algorithms' ability to predict injuries effectively. This analysis helps us understand how to configure datasets for better injury prediction, essential for improving sports injury prevention and player well-being.

The seven experiments were conducted with the aim of supporting our research objective, which is to investigate how state-of-the-art machine learning algorithms can predict and assist in reducing the risk of injuries in women's soccer. In the first experiment, we explored the potential of machine learning in helping coaches and team management identify the most important reasons for injury among soccer players. The second experiment focused on determining the most effective time frame for predicting injuries the following day. Our findings revealed the optimal time frame for injury prediction. In the third experiment, we have used eight different machine learning algorithms to predict injuries in women's soccer players. Notably, algorithms like XGBoost performed well in predicting injuries, while the Decision Tree algorithm showed moderate results. These experiments demonstrate the potential of machine learning in predicting injuries and supporting coaches and team management in their efforts to reduce the risk of injury. Experiment four is the extended version of experiment three where we have used hyperparameter tuning to enhance algorithm performance. Experiments five, six, and seven delved into different situations to further investigate the performance of our machine learning algorithms, providing a comprehensive understanding of our framework's capabilities in injury prediction.

Sports and technology have a significant impact on society, and soccer, being a widely popular sport, is no exception. With millions of fans and players across the globe, the sport has embraced the potential of machine learning techniques to enhance its efficiency. Machine learning has proven to be a valuable tool in soccer due to its ability to analyze vast amounts of data and make accurate predictions. Injuries are a major concern in soccer, affecting players, coaches, and team management. They not only have a personal impact on the players but can also be costly for teams in terms of points and potential championship opportunities. Our experimental findings are designed to assist soccer clubs and teams in identifying the factors closely associated with injuries. By leveraging our research, they can develop their own machine learning frameworks to predict and reduce the risk of injuries. Through our investigations, we have evaluated our machine learning framework in various scenarios, providing valuable insights and enabling researchers, soccer clubs, and teams to do further research and experiments.

5.2 Insights and Lessons Learned

This section presents the findings and lessons learned from our extensive experiments. Throughout our research, we conducted a total of seven experiments, which are all highlighted in this thesis. Below are some of the key insights obtained from our study.

- **Key Injury Correlates:** We have identified highly correlated factors with injuries that offer actionable insights. As discussed in Section 4.1. variables such as illness, weekly_load, ctl28, atl, ctl42, monotony, strain, and acwr emerged as significant predictors. This information is valuable for coaches and medical staff to proactively monitor and manage these risk factors. Understanding these correlations can aid in designing improved training and recovery protocols to mitigate injury risks.
- **Importance of Time Frame:** The selection of the time frame for injury prediction has a substantial impact on the performance of the algorithm. Our experiments showed that a 32-day prediction window is the most effective, balancing the need for timely predictions with the availability of sufficient historical data for accurate algorithm training. Shorter windows lacked adequate data for reliable predictions, while longer windows diluted the relevance of the information. This finding underscores the importance of selecting an appropriate time frame when designing predictive algorithms in sports contexts.
- **Algorithm Effectiveness and Selection:** One of the primary insights from our study is the varying effectiveness of different machine learning algorithms in predicting injuries. algorithms such as XGBoost and K-Nearest Neighbors demonstrated superior performance compared to others like Logistic Regression and Support Vector Machines. This suggests that algorithms capable of capturing complex, non-linear relationships in the data are more effective in this domain. This insight is crucial for practitioners looking to implement predictive analytics in sports, guiding them toward selecting more sophisticated ensemble methods for better results.
- **Hyperparameter Tuning:** We found that hyperparameter tuning significantly improved the performance of our machine-learning algorithms. By fine-tuning parameters such as the learning rate, the number of estimators, and other parameters, we achieved noticeable gains in the f1 score shown in Table 4.9. This underscores the importance of investing time and resources in optimizing algorithm parameters rather than relying on default settings. Practitioners should prioritize this step to enhance the predictive power of their algorithms.

- **General vs. Team-Specific algorithms:** In our study, we explored whether it is more effective to use a single algorithm for all teams or to develop separate algorithms for each team. Our results indicated that while a general algorithm provides reasonable predictions, team-specific algorithms offer a slight edge in accuracy. This improvement can be attributed to the unique characteristics and injury patterns of each team. Therefore, when resources allow, we recommend developing tailored algorithms for individual teams to achieve better predictive performance.
- **Evaluation on Injured-Player Datasets:** When we evaluated a dataset containing only injured players, the algorithms exhibited distinct performance characteristics. This evaluation underscored the importance of including a balanced representation of both injured and non-injured players in the training data to prevent biased predictions. It also demonstrated the algorithms' robustness in identifying injury-prone players, which is crucial for implementing early intervention strategies.
- **Dataset Scale and Performance:** In our further research, we found that the scale of the dataset had a considerable impact on algorithm performance. Larger datasets generally led to improved accuracy and stability of predictions, emphasizing the need for comprehensive data collection and integration efforts. However, we observed diminishing returns beyond a certain point, suggesting that while more data is beneficial, it is equally important to focus on the quality and relevance of the data.

In summary, our research highlights the importance of algorithm selection, appropriate time frames, hyperparameter tuning, and dataset considerations in predicting injuries using machine learning. These insights not only advance the academic understanding of sports injury prediction but also provide practical guidelines for implementing effective predictive analytics in women's soccer and potentially other sports.

5.3 Use Cases and Applications

Machine learning algorithms for injury prediction in sports, particularly women's soccer, offer diverse use cases and applications across various stakeholders involved in athlete management, sports performance optimization, and injury prevention. The following are some key use cases and applications of these algorithms:

- **Injury Risk Assessment:** Machine learning algorithms can assess the risk of injury for individual players based on their physiological, biomechanical, and performance data. By analyzing historical injury patterns and player-specific characteristics, these algorithms can identify athletes at higher risk of injury and provide early warning indicators to sports medicine professionals and coaching staff.
- **Training Load Management:** By integrating data from wearable sensors, training sessions, and player monitoring systems, machine learning algorithms can optimize training load management strategies to minimize the risk of overuse injuries and fatigue-related conditions. These algorithms can recommend personalized training programs tailored to individual player profiles, injury histories, and recovery status.
- **Game Strategy Optimization:** Machine learning algorithms can analyze match data, opponent profiles, and environmental factors to optimize game strategies that minimize

injury risk while maximizing performance outcomes. By identifying situational risk factors and player fatigue patterns, these algorithms can inform tactical decisions, substitution strategies, and game plans that prioritize player safety and well-being.

- **Rehabilitation Planning:** Following injury occurrence, machine learning algorithms can assist in designing personalized rehabilitation programs that promote optimal recovery and return-to-play timelines. By analyzing injury severity, recovery progress, and individual player characteristics, these algorithms can recommend targeted interventions, rehabilitation exercises, and progress monitoring protocols tailored to each athlete's specific needs.
- **Long-Term Injury Prevention:** Machine learning algorithms can contribute to the development of proactive injury prevention programs aimed at reducing the incidence and severity of injuries over the long term. By identifying modifiable risk factors, injury trends, and injury clusters within teams or player cohorts, these algorithms can inform targeted interventions, training modifications, and injury prevention strategies to enhance player durability and resilience.
- **Player Selection and Recruitment:** In talent identification and recruitment processes, machine learning algorithms can assist scouts and talent evaluators in assessing the injury risk and performance potential of prospective players. By analyzing player profiles, injury histories, and performance metrics, these algorithms can identify promising talents while considering their injury susceptibility and long-term athletic development prospects.
- **Public Health and Epidemiology:** Beyond individual athlete management, machine learning algorithms for injury prediction in women's soccer contribute to broader public health and epidemiological research efforts. By analyzing injury trends, risk factors, and injury mechanisms at the population level, these algorithms facilitate the development of evidence-based injury prevention policies, guidelines, and interventions that benefit the wider sports community.

In summary, machine learning algorithms for injury prediction in women's soccer have diverse applications spanning athlete health and performance management, coaching strategies, rehabilitation practices, talent identification, and public health initiatives. By leveraging advanced analytics and predictive algorithms techniques, these algorithms support data-driven decision-making and proactive approaches to injury prevention and athlete care across various domains within the sports industry.

5.4 Limitations

While developing machine learning algorithms to predict injuries in women's soccer, we encountered several limitations that need careful consideration. First, the quality and availability of data posed significant challenges. With a limited number of injury records compared to non-injury instances, the dataset was imbalanced, affecting the algorithms' accuracy. Moreover, dealing with missing data and integrating information from different sources added complexity to our analysis. Additionally, the dynamic nature of sports and athlete behavior posed challenges in capturing temporal changes and adapting algorithms accordingly. These limitations highlight the need for robust data collection, addressing imbalances, and continuous algorithm refinement to enhance accuracy and reliability in injury prediction for women's soccer.

5.5 Future Works

There are several promising directions we can explore to improve our machine-learning algorithms for predicting injuries in women's soccer. Here are some ideas for future work:

- **Surveys for Insight**: Conducting surveys with soccer teams can help us understand what type of performance forecasting would be most useful. These surveys can give us valuable feedback on how our algorithms can provide actionable data to help teams.
- **Focus on Objective Metrics**: We have seen the benefits of using GPS-derived features in our current work. However, we've also noticed inconsistencies with subjective wellness features. Therefore, focusing more on objective metrics like GPS measurements might lead to more accurate predictions. Forecasting these objective metrics could also yield new insights.
- **Online Learning and Deployment**: Implementing our algorithms in an online learning environment and deploying a passive machine learning analysis tool on the PmSys app could be the next step. This would allow us to continually update and improve our algorithms with new data.
- **Different Use Cases**: Instead of just focusing on wellness time series forecasting, we could explore other applications. For instance, predicting the movements or positions that lead to injuries or goals using GPS data could be beneficial.
- **Exploring Other Machine Learning Algorithms**: While we used a selection of machine learning algorithms for our study, trying out different algorithms might yield better results. We should experiment with algorithms outside our current selection.
- **algorithm Fine-Tuning**: Fine-tuning our algorithms with specific subsets of the data relevant to the players we are predicting could enhance accuracy. This approach allows for more personalized and accurate predictions.
- **Dynamic Parameter Selection**: By dynamically selecting the most relevant data configurations, such as input window size and features for each player and algorithm type, we can significantly improve our results.

There are many avenues to explore to further our understanding and application of machine learning in predicting athlete data. These methods can be applied not only to soccer but also to other sports to uncover important statistics and improve training conditions and strategies.

5.6 Recap of Contributions

Our contributions in this thesis span both computer science and sports science. We tackled the challenge of handling large amounts of missing data and class imbalance, providing detailed data preprocessing, manipulation, and feature engineering techniques. By analyzing features and addressing data gaps, we introduced effective preprocessing methods and engineered new features that enhance the performance of machine learning algorithms in predicting injuries. We identified key predictors and, based on these, developed additional features to improve algorithm accuracy. Moreover, we fine-tuned our machine learning algorithms to better handle the temporal nature of the dataset, making them more compatible for injury prediction. These efforts not only

advance the field of computer science by improving data handling techniques but also contribute to sports science by offering more accurate tools for injury prevention.

- **Research Contributions:** Our thesis paper and experimental findings make a significant contribution to the fields of machine learning and soccer. This research provides valuable insights and practical implications for predicting the risk of injuries in women's soccer.
- **Open Source Software:** We have built our machine learning framework to predict the injuries of women's soccer players. We have used Google Colab Pro to write our Python code. All the files are saved as IPYNB files and stored at the Simula PmSys GitHub repository (`https://github.com/simula/pmsys`).
- **Open Datasets:** We have created multiple datasets by using SoccerMon subjective metrics and extracting GPS features from the objective metrics. All the datasets are publicly available in the Simula Research Laboratory PmSys GitHub Repository (`https://github.com/simula/pmsys`).

5.7 Ethical Considerations

We need to think about the consequences of our research and the potential misuse and harm it can cause. Our algorithms have been affected by the huge imbalance in the data, leading to biases. This type of bias is one of the key issues in data science, as noted by Saltz et al. In our case, all the data is related to health or soccer performance and is meant to give important insights into each player involved. Therefore, it is crucial that our algorithms work properly, or they could harm the overall success and well-being of the team or individual players. We need to analyze the features and use feature importance metrics in real-world applications to understand how our algorithms weigh specific predictions.

Using readiness scores to choose players for important events might cause problems if players inaccurately report high readiness values to increase their chances of being chosen. This would lead to unreliable predictions because the data is not true. From a wellness perspective, this could make players fixate on an arbitrary number and cause stress, which goes against our goal of improving training conditions and game strategies. Therefore, our system needs to be used ethically, with consideration for the players' well-being. We should incorporate a trustworthy AI approach where each part can be explained and accounted for, especially when our predictions directly impact people.

In conclusion, the ethical implications of our research cannot be overlooked. By ensuring our algorithms are transparent, fair, and accurate, we can prevent potential harm and promote a healthier and more supportive environment for athletes. It's our responsibility to continue refining our methods, prioritizing the well-being of players, and fostering trust in our predictive tools. This commitment to ethical research practices will ultimately lead to better outcomes and a more positive impact on the world of sports.

5.8 Chapter Summary

In this chapter, we have synthesized the findings of our research and provided valuable insights into the prediction of injuries in women's soccer using machine learning techniques. We began

by revisiting the research questions outlined in Section 1.2. and addressing each question based on our empirical findings. Through rigorous analysis and experimentation, we have identified key predictors, optimized predictive algorithms, and evaluated their performance in injury prediction tasks.

Furthermore, we discussed the implications of our findings for sports science, injury prevention, and athlete well-being, highlighting the potential applications and use cases of predictive algorithms in women's soccer. We also acknowledged the limitations of our study, including data availability, algorithm generalizability, and ethical considerations, which warrant further investigation and consideration in future research endeavors.

Looking ahead, we outlined several avenues for future research, including the refinement of predictive algorithms, validation on larger and more diverse datasets, and the integration of advanced methodologies for causal inference and interpretability. By addressing these future directions, we aim to advance the field of injury prediction in women's soccer and contribute to the development of evidence-based strategies for athlete health and performance optimization.

In summary, this chapter consolidates the key findings, insights, and implications of our research, underscoring the importance of data-driven approaches in sports science and the potential of machine learning in enhancing injury prevention and athlete care in women's soccer. In the next chapter, we concluded our study by summarizing our findings, answering our sub-questions and main research objectives, and discussing future research directions shortly.

Chapter 6

Conclusion

In this thesis, we have explored the performance of machine learning algorithms to predict injuries in soccer players. We aimed to find the best methods to predict soccer player injury by evaluating different machine learning algorithms. Our findings include the potentiality and challenges of using machine learning in sports to solve real-life problems like injury prediction.

In the initial phase of our investigation, we focused on the subjective metrics of the SoccerMon dataset and extracted GPS features from the objective metrics. Our objective was to find out the features highly correlated with injury. This information might help coaches, team managers, and health professionals. After a thorough analysis, we discovered that features such as illness, weekly_load, ctl28, atl, ctl42, monotony, strain, and acwr have a high correlation with injury.

In our second experiment, we have introduced the window function to find out the best time frame to forecast injury for the following day. We have used a window size of 2, 4, 8, 16, and 32, where four algorithms (Decision Tree, K-Nearest Neighbors, LSTM, and XGBoost performed best with a time frame of 32 days. During the 16-day time frame, Logistic Regression, Random Forest, and Naive Bayes showed optimal results. Only Support Vector Machines performed best in an 8-day time frame, whereas no algorithm chose a 2- or 4-day time frame.

In our third experiment, we evaluated machine learning algorithms based on the F1 score keeping an eye on recall and precision scores. Each algorithm showed a different level of performance with its strengths and weaknesses. For example, the f1 score of Logistic regression is 0.26 showing moderate accuracy but having the possibility of improvement. The decision Tree algorithm scored 0.47 indicating a good balance of precision and recall. It makes the algorithm reliable in injury prediction. On the other hand, Random Forest scored 0.16 F1 score with 0.72 recall and 0.09 precision. The result shows the need to reduce false positives. On the other hand, K-Nearest Neighbors shows an F1 score of 0.34, where the recall and precision scores are 0.60 and 0.24, respectively. Like Random Forest, LSTM has shown a low F1 score of 0.18 with a recall and precision of 0.10 and 1.0. The Support Vector Machine performed a 0.27 F1 score with a balanced recall and precision. It is showing additional tuning. XGBoost performed 0.58 F1 score making it particularly effective in predicting injuries. Although Naive Bayes scored 0.12 F1 score with high recall and low precision showing the need for further tuning. Lastly, the Naive Bayes performed poorly, with an F1 score of 0.12. After performing hyperparameter tuning, among these seven machine learning algorithms, Logistic Regression, Random Forest, K-Nearest Neighbors, Support Vector Machine, and Naive Bayes showed better performance, whereas the performance of Decision Tree and

XGBoost did not improved.

By focusing on a specific team (Team A), we obtained a more consistent dataset that allowed us to identify specific injury patterns. However, this approach may limit the generalizability of our findings. For Team A, K-Nearest Neighbors performed very well with an F1 score of 0.50. In contrast, LSTM, with a high recall (0.80) but low precision (0.08), had an F1 score of 0.14, indicating a significant number of false positives. We also analyzed our machine learning algorithm's performance on only injured players, but this method may limit its ability to predict new injuries. For example, XGBoost performed excellently with an F1 score of 0.58, and the Decision Tree showed a moderate performance with an F1 score of 0.35. Overall, each approach had its advantages and limitations, highlighting the complexities of using machine learning for injury prediction in sports.

In our last experiment, we explored the impact of dataset size on our machine learning algorithms. For the full dataset, we have struggled with class imbalance because of fewer injured records, which could bias algorithms in predicting non-injury cases. Among the eight algorithms, XGBoost performed considerably, with an F1 score of 0.58. When we moved forward and analyzed the algorithm performance for only a specific team (Team A), we explored that K-Nearest Neighbors showed the most promising result with an F1 score of 0.50 and XGBoost of 0.40. Lastly, we made a dataset focused on only injured players, where XGBoost performed best with an F1 score of 0.58 and recall and precision of 0.50 and 0.71, but these results could be biased as they only focused on injured players.

Our research highlights the diverse applications of machine learning algorithms in sports, including injury risk assessment, training load management, game strategy optimization, rehabilitation planning, long-term injury prevention, player selection, and public health. These applications underscore the potential for data-driven approaches to enhance athlete health and performance.

Despite the promising results, our study faced several limitations, such as data quality and availability, the complexity of predicting injuries, and algorithm interpretability. Addressing these limitations is crucial to improving the accuracy and effectiveness of machine learning algorithms in sports injury prediction.

The insights of this study would be useful for sports practitioners, coaches, and medical professionals in practical implications. Using machine learning-based injury prediction algorithms, they can make important decisions in their training programs, and tactical planning, and manage team formation proactively.

For future research, we recommend integrating multi-modal data, conducting longitudinal studies, exploring advanced algorithm architectures, leveraging transfer learning, implementing real-time injury surveillance systems, and developing ethical frameworks for predictive analytics in sports injury management. By addressing these areas, we can further refine predictive algorithms and develop robust, ethical, and effective injury prevention strategies, leading to a safer and more sustainable sports environment.

Bibliography

[1] J. Gualtieri and S. Chettri, "Support vector machines for classification of hyperspectral data," in *IGARSS 2000. IEEE 2000 International Geoscience and Remote Sensing Symposium. Taking the Pulse of the Planet: The Role of Remote Sensing in Managing the Environment. Proceedings (Cat. No. 00CH37120)*, IEEE, vol. 2, 2000, pp. 813–815.

[2] A. J. Noblet and S. M. Gifford, "The sources of stress experienced by professional australian footballers," *Journal of applied sport psychology*, vol. 14, no. 1, pp. 1–13, 2002.

[3] C. Finch, "A new framework for research leading to sports injury prevention," *Journal of science and medicine in sport*, vol. 9, no. 1-2, pp. 3–9, 2006.

[4] C. W. Fuller, J. Ekstrand, A. Junge, *et al.*, "Consensus statement on injury definitions and data collection procedures in studies of football (soccer) injuries," *Scandinavian journal of medicine & science in sports*, vol. 16, no. 2, pp. 83–92, 2006.

[5] S. A. Neslin, S. Gupta, W. Kamakura, J. Lu, and C. H. Mason, "Defection detection: Measuring and understanding the predictive accuracy of customer churn models," *Journal of marketing research*, vol. 43, no. 2, pp. 204–211, 2006.

[6] R. Witzig, *The global art of soccer*. CusiBoy Publishing, 2006.

[7] A. Junge and J. Dvorak, "Injuries in female football players in top-level international tournaments," *British journal of sports medicine*, vol. 41, no. suppl 1, pp. i3–i7, 2007.

[8] A. H. Engebretsen, G. Myklebust, I. Holme, L. Engebretsen, and R. Bahr, "Prevention of injuries among male soccer players: A prospective, randomized intervention study targeting players with previous injuries or reduced function," *The American journal of sports medicine*, vol. 36, no. 6, pp. 1052–1060, 2008.

[9] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, Ieee, 2008, pp. 1322–1328.

[10] K. Steffen, H. Bakka, G. Myklebust, and R. Bahr, "Performance aspects of an injury prevention program: A ten-week intervention in adolescent female football players," *Scandinavian journal of medicine & science in sports*, vol. 18, no. 5, pp. 596–604, 2008.

[11] M. Ghazanfar and A. Prugel-Bennett, "An improved switching hybrid recommender system using naive bayes classifier and collaborative filtering," 2010.

[12] B. Clarsen, O. Rønsen, G. Myklebust, T. W. Flørenes, and R. Bahr, "The oslo sports trauma research center questionnaire on health problems: A new approach to prospective monitoring of illness and injury in elite athletes," *British journal of sports medicine*, vol. 48, no. 9, pp. 754–760, 2014.

[13] N. Datson, A. Hulton, H. Andersson, *et al.*, "Applied physiology of female soccer: An update," *Sports medicine*, vol. 44, pp. 1225–1240, 2014.

[14] T. T. Hoang, "Pmsys: Implementation of a digital player monitoring system," M.S. thesis, 2015.

[15] D. C. MĂNESCU, "Elements of the specific conditioning in football at universitary level," *Editura ASE*, 2015.

[16] M. K. Drew and C. F. Finch, "The relationship between training load and injury, illness and soreness: A systematic and literature review," *Sports medicine*, vol. 46, pp. 861–883, 2016.

[17] T. J. Gabbett, "The training—injury prevention paradox: Should athletes be training smarter and harder?" *British journal of sports medicine*, vol. 50, no. 5, pp. 273–280, 2016.

[18] Y. Zhong, "The analysis of cases based on decision tree," in *2016 7th IEEE international conference on software engineering and service science (ICSESS)*, IEEE, 2016, pp. 142–147.

[19] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons. b*, vol. 4, pp. 51–62, 2017.

[20] M. S. Shelke, P. R. Deshmukh, and V. K. Shandilya, "A review on imbalanced data handling using undersampling and oversampling technique," *Int. J. Recent Trends Eng. Res*, vol. 3, no. 4, pp. 444–449, 2017.

[21] A. J. Wyner, M. Olson, J. Bleich, and D. Mease, "Explaining the success of adaboost and random forests as interpolating classifiers," *Journal of Machine Learning Research*, vol. 18, no. 48, pp. 1–33, 2017.

[22] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for knn classification," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 3, pp. 1–19, 2017.

[23] S. Bridgewater, "Women and football," in *Routledge handbook of football business and management*, Routledge Abingdon, 2018, pp. 351–365.

[24] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

[25] K. Kirasich, T. Smith, and B. Sadler, "Random forest vs logistic regression: Binary classification for heterogeneous datasets," *SMU Data Science Review*, vol. 1, no. 3, p. 9, 2018.

[26] C. Moseid, G. Myklebust, M. Fagerland, B. Clarsen, and R. Bahr, "The prevalence and severity of health problems in youth elite sports: A 6-month prospective cohort study of 320 athletes," *Scandinavian journal of medicine & science in sports*, vol. 28, no. 4, pp. 1412–1423, 2018.

[27] S. A. Pettersen, H. D. Johansen, I. A. Baptista, P. Halvorsen, and D. Johansen, "Quantified soccer using positional data: A case study," *Frontiers in physiology*, vol. 9, p. 314 255, 2018.

[28] J. G. Claudino, D. d. O. Capanema, T. V. de Souza, J. C. Serrão, A. C. Machado Pereira, and G. P. Nassis, "Current approaches to the use of artificial intelligence for injury risk assessment and performance prediction in team sports: A systematic review," *Sports medicine-open*, vol. 5, pp. 1–12, 2019.

[29] P.-J. Lin, S. Hung, S. F. S. Lam, and B. C. Chen, "Object recognition with machine learning: Case study of demand-responsive service," in *2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, IEEE, 2019, pp. 129–134.

[30] A. B. Shaik and S. Srinivasan, "A brief survey on random forest ensembles in classification model," in *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Volume 2*, Springer, 2019, pp. 253–260.

[31] S. S. Bafjaish, "Comparative analysis of naive bayesian techniques in health-related for classification task," *Journal of Soft Computing and Data Mining*, vol. 1, no. 2, pp. 1–10, 2020.

[32] K. M. Crossley, B. E. Patterson, A. G. Culvenor, A. M. Bruder, A. B. Mosler, and B. F. Mentiplay, "Making football safer for women: A systematic review and meta-analysis of injury prevention programmes in 11 773 female football (soccer) players," *British journal of sports medicine*, vol. 54, no. 18, pp. 1089–1098, 2020.

[33] E. Eliakim, E. Morgulev, R. Lidor, and Y. Meckel, "Estimation of injury costs: Financial damage of english premier league teams' underachievement due to injuries," *BMJ Open Sport & Exercise Medicine*, vol. 6, no. 1, e000675, 2020.

[34] A. Griffin, I. C. Kenny, T. M. Comyns, and M. Lyons, "The association between the acute: Chronic workload ratio and injury and its application in team sports: A systematic review," *Sports Medicine*, vol. 50, pp. 561–580, 2020.

[35] F. M. Impellizzeri, M. S. Tenan, T. Kempton, A. Novak, and A. J. Coutts, "Acute: Chronic workload ratio: Conceptual issues and fundamental pitfalls," *International journal of sports physiology and performance*, vol. 15, no. 6, pp. 907–913, 2020.

[36] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine learning with oversampling and undersampling techniques: Overview study and experimental results," in *2020 11th international conference on information and communication systems (ICICS)*, IEEE, 2020, pp. 243–248.

[37] I. H. Sarker, A. Colman, J. Han, A. I. Khan, Y. B. Abushark, and K. Salah, "Behavdt: A behavioral decision tree learning to build user-centric context-aware predictive model," *Mobile Networks and Applications*, vol. 25, pp. 1151–1161, 2020.

[38] A. Viloria, O. B. P. Lezama, and N. Mercado-Caruzo, "Unbalanced data processing using oversampling: Machine learning," *Procedia Computer Science*, vol. 175, pp. 108–113, 2020.

[39] C. Wang, J. T. Vargas, T. Stokes, R. Steele, and I. Shrier, "Analyzing activity and injury: Lessons learned from the acute: Chronic workload ratio," *Sports Medicine*, vol. 50, no. 7, pp. 1243–1254, 2020.

[40] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, "A survey on long short-term memory networks for time series prediction," *Procedia Cirp*, vol. 99, pp. 650–655, 2021.

[41] C. B. Vennerød, A. Kjærran, and E. S. Bugge, "Long short-term memory rnn," *arXiv preprint arXiv:2105.06756*, 2021.

[42] S. Kulakou, N. Ragab, C. Midoglu, *et al.*, "Exploration of different time series models for soccer athlete performance prediction," *Engineering Proceedings*, vol. 18, no. 1, p. 37, 2022.

[43] E. D. McAlpine, P. Michelow, and T. Celik, "The utility of unsupervised machine learning in anatomic pathology," *American Journal of Clinical Pathology*, vol. 157, no. 1, pp. 5–14, 2022.

[44] N. Ragab, "Soccer athlete performance prediction using time series analysis," M.S. thesis, OsloMet-storbyuniversitetet, 2022.

[45] A. K. Tyagi and P. Chahal, "Artificial intelligence and machine learning algorithms," in *Research Anthology on Machine Learning Techniques, Methods, and Applications*, IGI Global, 2022, pp. 421–446.

[46] F. Aiello, F. M. Impellizzeri, S. J. Brown, A. Serner, and A. McCall, "Injury-inciting activities in male and female football players: A systematic review," *Sports medicine*, vol. 53, no. 1, pp. 151–176, 2023.

[47] Z. A. Ali, Z. H. Abduljabbar, H. A. Taher, A. B. Sallow, and S. M. Almufti, "Exploring the power of extreme gradient boosting algorithm in machine learning: A review," *Academic Journal of Nawroz University*, vol. 12, no. 2, pp. 320–334, 2023.

[48] M. Boeker and C. Midoglu, "Soccer athlete data visualization and analysis with an interactive dashboard," in *International Conference on Multimedia Modeling*, Springer, 2023, pp. 565–576.

[49] A. Culvin and A. Bowes, "Introduction: Women's football in a global, professional era," in *Women's Football in a Global, Professional Era*, Emerald Publishing Limited, 2023, pp. 1–13.

[50] L. Hoel, *UiO.2023*, 2023. [Online]. Available: https://home.simula.no/~paalh/students/LarsHoel-UiO.2023.pdf.

[51] J. K. Vollstad, "Developing a linear mixed model to predict rpe and srpe in female elite football players using external load measures," M.S. thesis, UiT The Arctic University of Norway, 2023.

[52] C. Midoglu, A. Kjæreng Winther, M. Boeker, *et al.*, "A large-scale multivariate soccer athlete health, performance, and position monitoring dataset," *Scientific Data*, vol. 11, no. 1, p. 553, 2024.

[53] N. Elliot and F. Onuodu, "The role of artificial intelligence (ai) in the near future,"

[54] Howden, *Howden's European Football Injury Index reveals record injury cost of over £500m for 2021/22 season*. [Online]. Available: https://www.howdengroup.com/news-and-insights/howdens-european-footbal-Injury-Index-reveals-record-injury-cost-of-over-500m-for-2021-22-season.

[55] A. L. Jarmann, *Identifying Injury Risk Factors for Elite Soccer Teams Using Survival Analysis*. [Online]. Available: https://home.simula.no/~paalh/students/AnnaLJarmann-UiO-2023.pdf.

[56] J. F. Lin, *Football training*. [Online]. Available: https://unsplash.com/@jeffreyflin.

[57] C. Midoglu, *SoccerMon: A Large-Scale Multivariate Soccer Athlete Health, Performance, and Position Monitoring Dataset*. [Online]. Available: https://zenodo.org/records/10033832.

[58] N. Ragab, *Soccer Athlete Performance Prediction using Time Series Analysis*. [Online]. Available: https://home.simula.no/~paalh/students/NourhanRagab-OsloMet-2022.pdf.

[59] M. M. Sagbakken, *Using Machine Learning to Predict Elite Female Athletes' Readiness to Play in Soccer*. [Online]. Available: https://home.simula.no/~paalh/students/MathiasMSagbakken-UiO-2023.pdf.

[60] N. A.-R. Al-Serw, *Undersampling and oversampling: An old and a new approach*. [Online]. Available: https://medium.com/analytics-vidhya/undersampling-and-oversampling-an-old-and-a-new-approach-4f984a0e8392.

[61] StatSports, *WOMEN'S APEX ATHLETE SERIES - GPS PERFORMANCE TRACKER*. [Online]. Available: https://uk.shop.statsports.com/products/ladies-apex-athlete-series-gps-performance-tracker.

Appendix A

Additional Figures

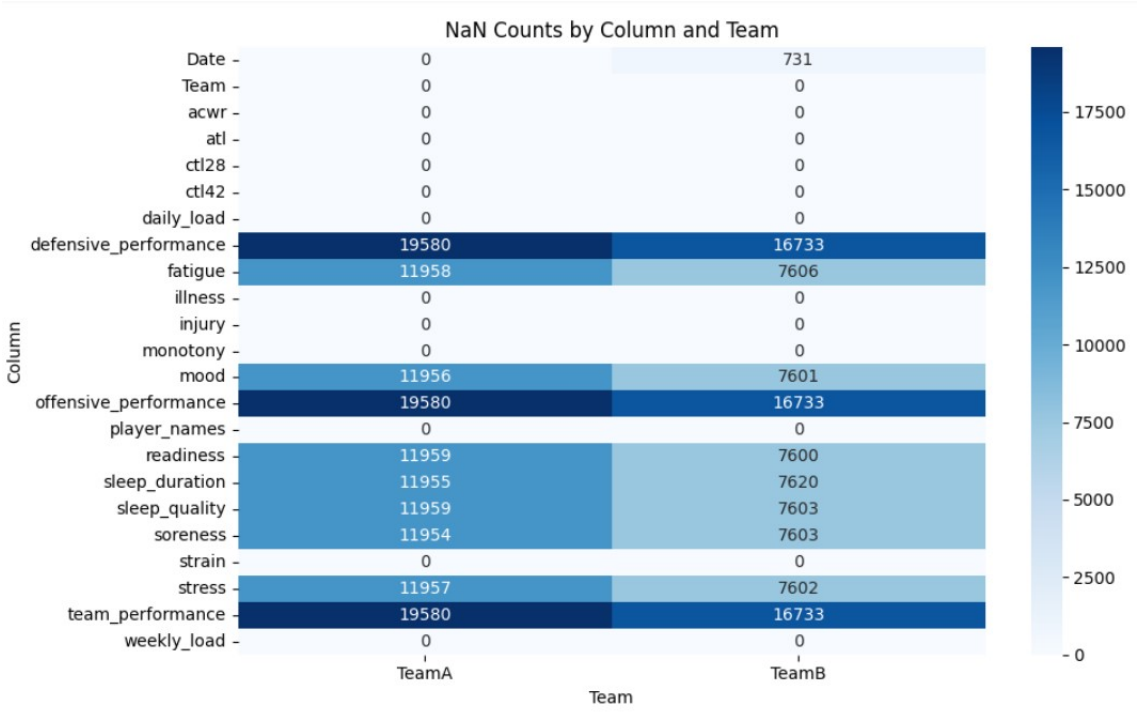

Figure A.1: The figure illustrates the presence of NaN values for both teams.

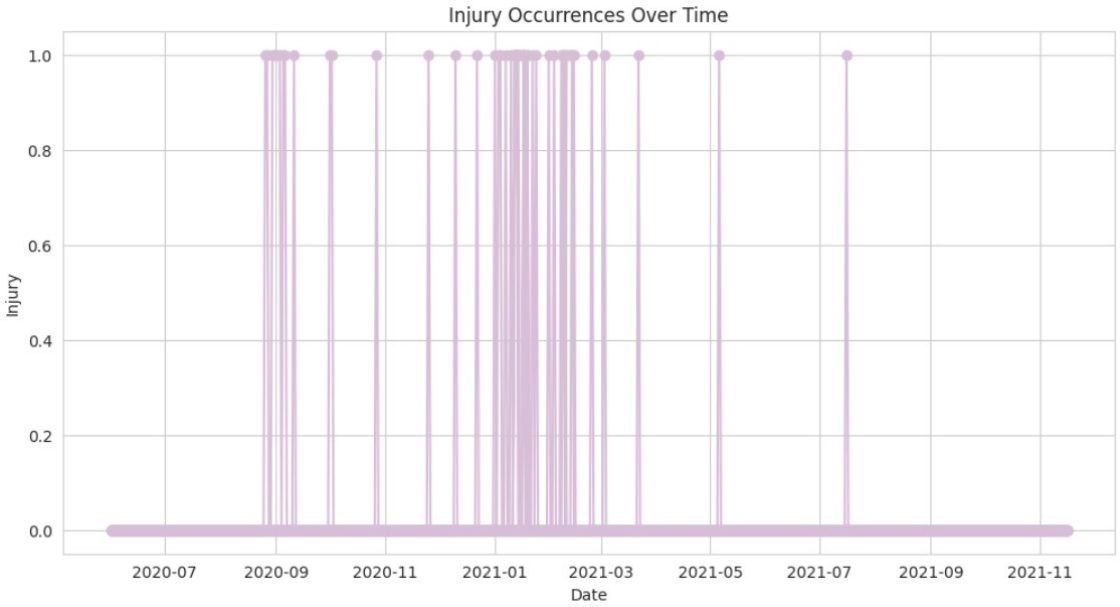

Figure A.2: Injury records of soccer players from July 2020 to November 2021.

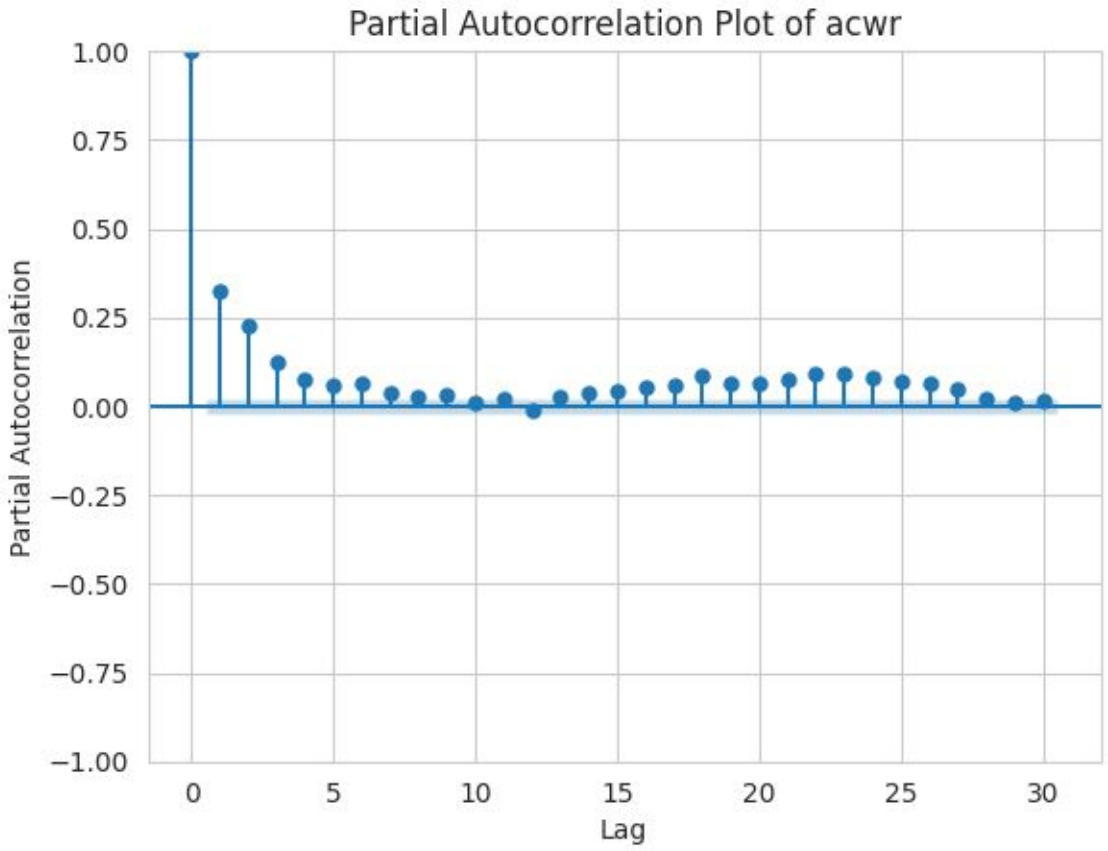

Figure A.3: Partial autocorrelation of acwr.

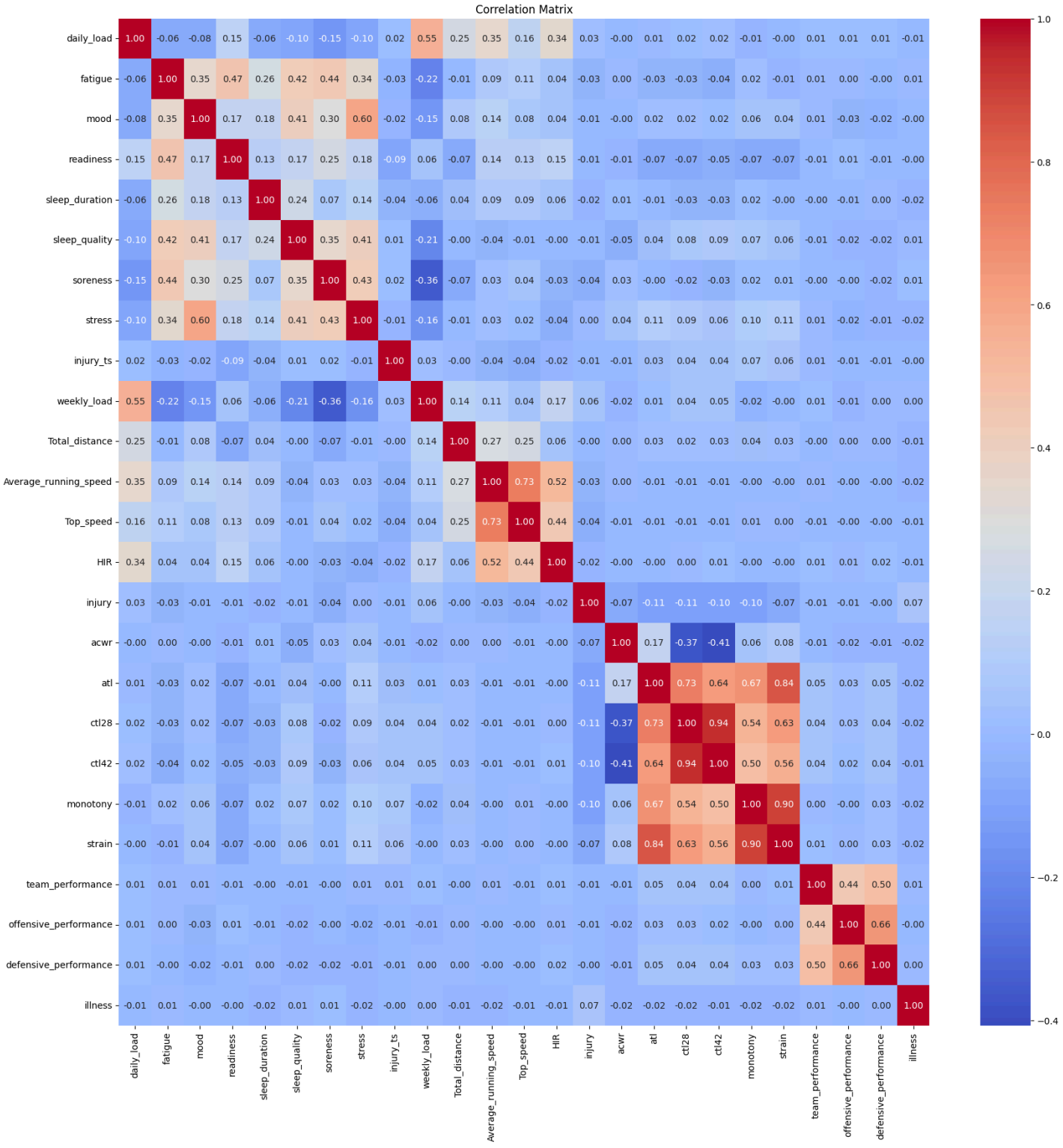

Figure A.4: The correlation matrix heatmap visualizes the relationships between various performance and wellness variables, with color intensity indicating the strength of correlations.

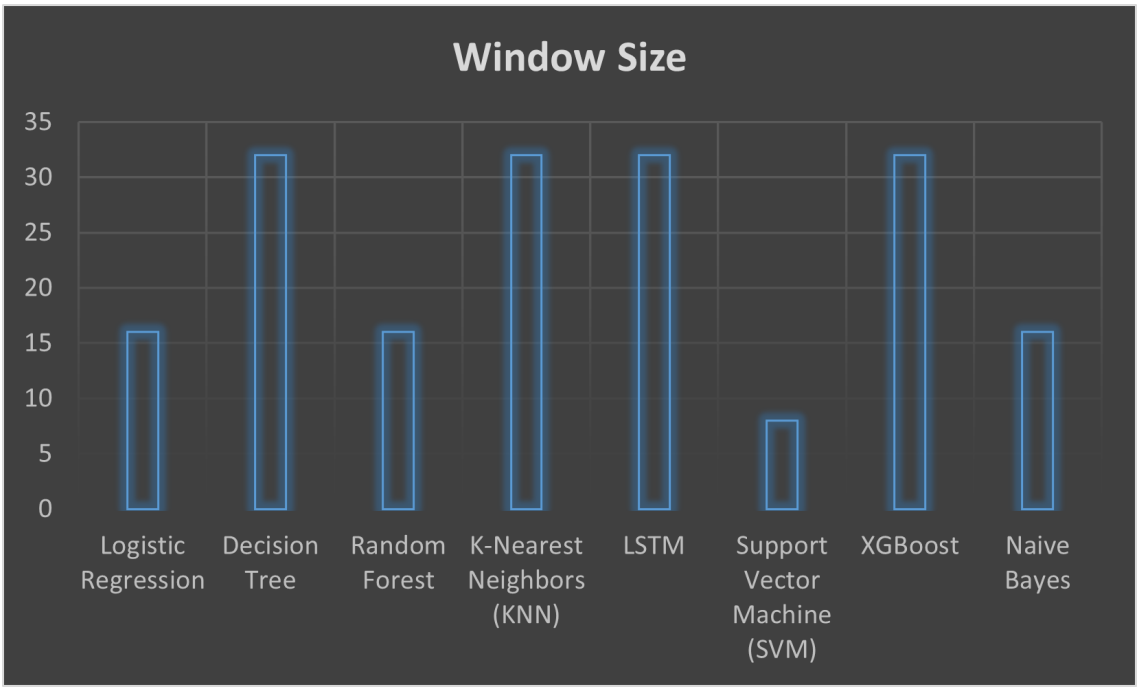

Figure A.5: The bar chart represents information about the window size preference of different algorithms.

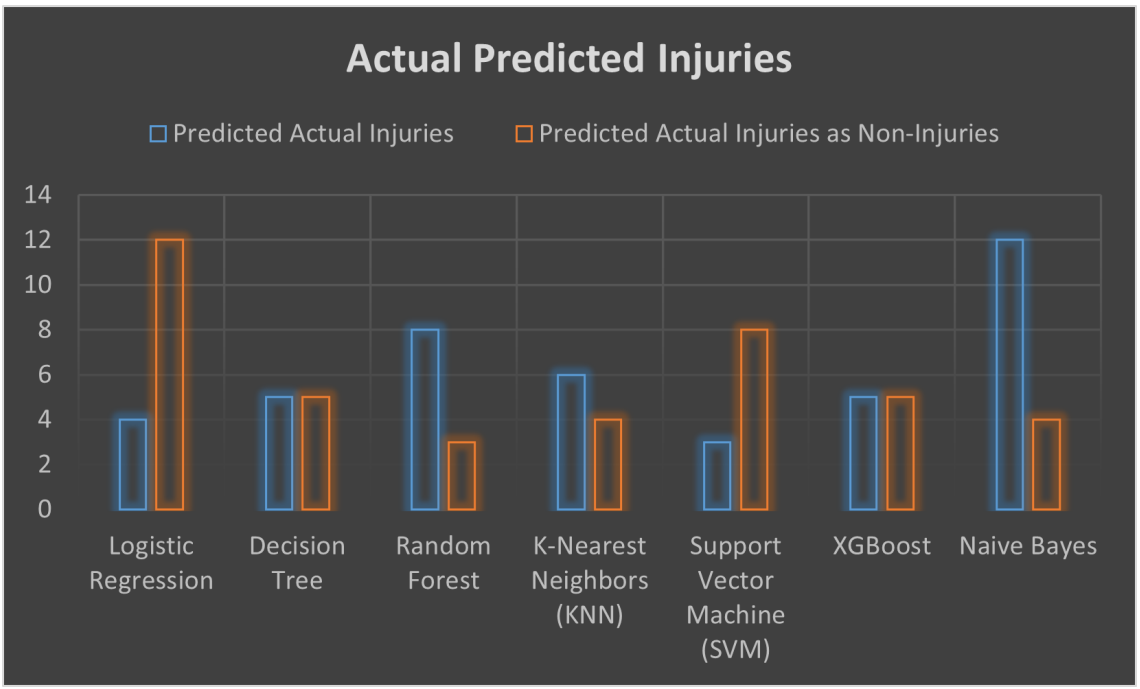

Figure A.6: The bar charts show information about algorithms that predict actual injuries and actual injuries predicted as non-injuries.

Appendix B

List of Abbreviations

- **ALT** - Acute Training Load.
- **KNN** - K-Nearest Neighbors.
- **ADASYN** - Adaptive Synthetic Sampling.
- **LSTM** - Long Short Term Memory.
- **SVM** - Support Vector Machine.
- **SMOTE** - Synthetic Minority Over-sampling Technique.
- **PmSys** - Player Monitoring System.
- **ACWR** - Acute:Chronic Workload Ratio.
- **sRPR** - Session Rating of Perceived Exertion.
- **CTL28** - Chronic Training Load over the past 28 days.
- **CTL 42** - Chronic Training Load over the past 42 days.
- **HIR** - High Intensity Running.
- **RNN** - Recurrent Neural Network.

Appendix C

Additional Tables

Models	Input Windows	Recall	Precision	F1	Accuracy
Logistic Regression	16	0.25	0.33	0.28	0.99
Decision Tree	32	0.40	0.57	0.47	0.99
Random Forest	16	0.72	0.11	0.20	0.96
K-Nearest Neighbor(KNN)	32	0.30	1.0	0.46	0.99
Support Vector Machine (SVM)	16	0.18	0.60	0.28	0.99
XGBoost	32	0.40	1.0	0.57	0.99
Naive Bayes	32	0.60	0.09	0.16	0.96

Table C.1: Algorithms performance using different window sizes.

Østfold University College
Department of Computer Science and Communication

Agreement Between Simula and Østfold University College:
Master Thesis for Mohaiminul Islam Emon

Master student Mohaiminul Islam Emon at Østfold University College (ØUC) will write his master thesis (giving 60 ECTS credits) in Applied Computer Science in the Academic Year 2023/2024. The title of the master project, that will be hosted externally by Simula Research/Metropolitan, is:

"Collection and AI-Based Analysis of Athlete Training and Wellness Data"*

Simula and ØUC agree on the following regarding this master project:

1. The supervision will be shared between the two external supervisors at Simula:
   - Professor Chief Research Scientist/Research Professor Pål Halvorsen
   - Chief Research Scientist/Research Professor Michael Riegler,
   - Postdoctoral Fellow Cise Midoglu

   and the internal supervisor at ØUC:

   - Accociate Professor Lars Vidar Magnusson.
2. The student will have access to master labs both at Simula and ØUC.
3. Results from the master project may be published openly with credits to both institutions.
4. All rules and regulations for a master thesis in Applied Computer Science at ØUC apply†.
5. The grading of the master thesis will be done by the internal supervisor at ØUC and an additional external evaluator appointed by ØUC.
6. The student's oral examination/defense of the master thesis will be arranged and conducted by ØUC. The defense will take place in Halden, participants may also join digitally.

This agreement has been sent to the student, to the internal and external supervisors and to the head of the Department of Computer Science and Communication. All involved parties have accepted the agreement.

Halden. June 2023

/S/ Jan Høiberg

Jan Høiberg
Coordinator for the Master Program in Applied Computer Science

*: Project description: https://www.simula.no/education/studies/collection-and-ai-based-analysis-athlete-training-and-wellness-data
†: Master thesis course description: https://www.hiof.no/english/studies/courses/iio/itk/2023/autumn/iti52020.html

simulamet

Pilestredet 52 | 0167 Oslo | Norway
post@simula.no | www.simulamet.no

Visiting address:
Pilestredet 52, Oslo, Norway

15. juni 2024

Nondisclosure Agreement

This Nondisclosure Agreement or (”Agreement”) has been entered into on the date of Tuesday, 27 June 2023, and is by and between:

- Party Disclosing Information (“Disclosing Party”):
  Simula Metropolitan Center for Digital Engineering (SimulaMet)
  with a mailing address of Pilestredet 52, 0167 Oslo, Norway.
- Party Receiving Information (“Receiving Party”):
  Md Mohaiminul Islam Emon
  with a mailing address of Sandakerveien 76 A, 0484 Oslo, Norway.

For the purpose of preventing the unauthorized disclosure of Confidential Information as defined below. The parties agree to enter into a confidential relationship concerning the disclosure of certain proprietary and confidential information (”Confidential Information”).

1. Definition of Confidential Information. For purposes of this Agreement, ”Confidential Informationsshall include all information or material that (i) has or could have commercial value or other utility in the business in which Disclosing Party is engaged, (ii) is potentially subject to a data handling plan, which aims to ensure good and safe handling of data throughout the research process. Confidential Information includes but is not limited to datasets which pertain to athlete injuries, collected from soccer teams in relation to injury research conducted by the Disclosing Party, as well as datasets which pertain to soccer athlete performance, from third parties. Confidential Information transmitted in written form shall remain only on the computation servers hosted by the Disclosing Party, to which the Receiving Party shall be provided with access.

2. Exclusions from Confidential Information. Receiving Party’s obligations under this Agreement do not extend to information that is: (a) publicly known at the time of disclosure or subsequently becomes publicly known through no fault of the Receiving Party; (b) discovered or created by the Receiving Party before disclosure by Disclosing Party; or (c) learned by the Receiving Party through legitimate means other than from the Disclosing Party or Disclosing Party’s representatives.

3. Obligations of Receiving Party. Receiving Party shall hold and maintain the Confidential Information in strictest confidence for the sole and exclusive benefit of the Disclosing Party. Receiving Party shall not, without the prior written approval of Disclosing Party, use for Receiving Party’s benefit, publish, copy, or otherwise disclose to others, or permit the use by others for their benefit or to the detriment of Disclosing Party, any Confidential Information. Receiving Party shall return to Disclosing Party any and all records, notes, and other written, printed, or tangible materials in its possession pertaining to Confidential Information immediately if Disclosing Party requests it in writing.

4. Time Periods. The nondisclosure provisions of this Agreement shall survive the termination of this Agreement and Receiving Party’s duty to hold Confidential Information in confidence shall remain in effect until the Confidential Information no longer qualifies as confidential (i.e. is made public), or until Disclosing Party sends Receiving Party written notice releasing Receiving Party from this Agreement, whichever occurs first.

5. Relationships. Nothing contained in this Agreement shall be deemed to constitute either party a partner, joint venture or employee of the other party for any purpose.

6. Severability. If a court finds any provision of this Agreement invalid or unenforceable, the remainder of this Agreement shall be interpreted so as best to affect the intent of the parties.

7. Integration. This Agreement expresses the complete understanding of the parties with respect to the subject matter and supersedes all prior proposals, agreements, representations, and understandings. This Agreement may not be amended except in writing signed by both parties.

8. Waiver. The failure to exercise any right provided in this Agreement shall not be a waiver of prior or subsequent rights.

9. Notice of Immunity. Receiving Party shall not be held criminally or civilly liable under any federal or state trade secret law for the disclosure of Confidential Information that is made (i) in confidence to a federal, state, or local government official, either directly or indirectly, or to an attorney; and (ii) solely for the purpose of reporting or investigating a suspected violation of law; or is made in a complaint or other document filed in a lawsuit or other proceeding, if such filing is made under seal.

10. Purpose and Duration of the Provision The Disclosing Party acknowledges that the Confidential Information provided to the Receiving Party is for the sole purpose of conducting research within the context of a Master's Thesis. The provision shall commence on the effective date of this Agreement and shall continue until the completion of the Master's Thesis.

This Agreement and each party's obligations shall be binding on the representatives, assigns and successors of such party. Each party has signed this Agreement through its authorized representative.

DISCLOSING PARTY

Typed or Printed Name: Pål Halvorsen (as the authorized representative of SimulaMet)
Signature:

Mohaiminul Islam

RECEIVING PARTY

Typed or Printed Name: Md Mohaiminul Islam Emon
Signature: