Soccer Athlete Performance Prediction using Time Series Analysis

Nourhan Ragab

Thesis submitted for the degree of
Master in Applied Computer and Information Technology (ACIT)
30 credits

Department of Computer Science
Faculty of Technology, Art and Design

OSLO METROPOLITAN UNIVERSITY

Spring 2022

Soccer Athlete Performance Prediction using Time Series Analysis

Nourhan Ragab

Soccer Athlete Performance Prediction using Time Series Analysis

http://www.oslomet.no/

Printed: Oslo Metropolitan University

Abstract

Regardless of the sport you prefer, your favorite athlete has almost certainly disappointed you at some point. Did you jump to a conclusion and dismissed it as "not their day"? Or, did you consider the underlying causes for their poor performance on that particular day? Under-performance can have big consequences in team sports such as soccer and affect the entire team dynamic. Basal needs like sleep quality and wellness parameters such as mood, fatigue and muscle soreness can affect an athlete's performance. In this context, the practice of using wearable sensor devices to quantify athlete health and performance is gaining popularity in sports science. This thesis aims to predict how ready a soccer athlete is to train or play a match based on the subjectively reported wellness parameter *readiness to play*, collected by the PMSys athlete performance monitoring system [34, 33, 17]. Even though women's soccer is receiving increasingly more attention, with a recent record in game day attendance marking over 90.000 spectators [50], the vast majority of soccer studies are conducted on male athletes. In this sense, we explore a relatively new domain using the PMSys dataset, which is from two Norwegian elite female soccer clubs over the period of 2020 and 2021. We predict readiness by utilizing the Long short-term memory (LSTM) method and the Tsai [45] state-of-the-art deep learning library. We develop a framework that is able to handle univariate multistep time series prediction and easily allows for further development. The experimental results show that it is possible to train a Machine Learning (ML) model on a team and predict a single player's readiness, detecting detect peaks closely to actual values. It is possible to use the previous week to predict the upcoming day, or even the upcoming week, as the model does not require much data to get started. The model works well on data from the entire team for a shorter period than a larger set of data for a longer period, which allows the teams to quickly start using the system with existing data. Hyperparameters are easily configurable and can be changed as required to optimize the model. Our results can be used for evidence based decisions, such as benching the team star so she doesn't get injured for the rest of the season. As a first milestone, this framework will be incorporated in PMSys and used in the Norwegian the elite female soccer league, Toppserien, but the overall approach can be part of a standardized athlete performance monitoring system that is globally used by athletes in all sports.

Keywords— Machine Learning (ML), Artificial Intelligence (AI), Long short-term memory (LSTM), Univariate, Deep Learning, Time Series Prediction, Female Soccer, Tsai, Python, Amazon Web Services (AWS)

Acknowledgments

First and foremost, I would like to express my gratitude towards my supervisors Cise Midoglu, Michael Riegler and Pål Halvorsen for invaluable guidance and the opportunity to work with this assignment.

I also want to express my gratitude to my closest friend Anita Ilieva, who always found time to assist and encourage me throughout my degree and thesis work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Team sports are gaining traction with association soccer (soccer) in the lead as the most watched sport on television [18]. In 2018, 3.572 billion viewers tuned in to watch the FIFA World Cup [10]. This is equivalent to half of the population on the globe [10]. Soccer is played by amateurs and professionals globally and is a sport that brings people together across the world.

Despite the fact that soccer has traditionally been a male-dominated sport, interest in female soccer has grown significantly in the last 10 years. This is especially true in terms of match-day attendance and TV audience [38]. However, research on women's soccer is still insufficient. This must change in order to tailor the intensity and quantity of training to meet the demands of women and prevent injuries [19].

For most professional sports achievements, athletes' individual physical abilities are combined with many modern technologies from fields such as medicine, equipment production, nutrition, and physical and mental health monitoring. Proper diet, rest, and training regimens, as well as continuous monitoring and analysis of wellness and performance metrics can make a big difference for both individual and team sports. Soccer players constantly adhere to a strict nutrition plan, training process, and rest regime so that their body is in the required state at particular moments. The athletes' condition is influenced not only by the amount of consumed and burned calories, or the duration and intensity of the training process, but also by parameters such as the duration and quality of their sleep and their general mental state including mood and stress level.

The increasing popularity and adoption of ML approaches has led to more evidence-based decision making [35]. In this context, it is possible to compile a set of parameters describing the general state of the athlete at a certain point of time, collect objective or subjective measurements of these parameters, and try to predict the state/behavior of the athlete's body in the near future using ML. For instance, according to Fuller et al. [12], an average soccer team can expect around 50 injuries per season. Using ML, it might be feasible to make evidence-based decisions for reducing injuries at a chosen period in the future using time series analysis and predictions.

The desired outcome of implementing such technologies into sports science is that injuries will be reduced, performance will be improved, and better decisions will be made. Similar technologies have already been approved by organizations like Fédération Internationale de Football Association (FIFA) and are used by several teams in the Norwegian top league. In this thesis, we will use a

commercially deployed digital system for soccer athlete monitoring built on a cloud-based microservice architecture called PMSys [34].

1.2 Problem Statement

As mentioned above, the continuous collection of metrics from training sessions and matches allows for data analysis and visualization, allowing soccer teams to monitor and improve overall athlete performance and avoid injuries. One of the most important data analysis use cases is forecasting (prediction).

In this thesis, we will focus on one of the wellness parameters collected by PMSys [17], *readiness to play*, which is a measure of a player's ability to complete a training session or play in a match. Based on the subjective self-reported daily reports of readiness by athletes, we will address the problem of predicting soccer players' ability to perform. Our ultimate goal will be to suggest how such a framework can be implemented in existing cloud-based distributed pipeline systems, to provide both coaches and developers with continuous feedback. Our overall research question is as follows:

> *Can we predict readiness for elite female soccer athletes using machine learning on data collected using an athlete monitoring system?*

This research question is further broken down into several sub-questions:

RQ1. Is it more accurate to predict a player's readiness using data from an individual player or team based data?

RQ2. How far back in the historical data should we go and how far into the future can we predict?

RQ3. Does the training dataset size have an impact on the results, and is a year or two the best for accurate predictions?

RQ4. What permutation of the hyperparameters results in more accurate predictions?

1.3 Scope

The goal of this thesis is to use time series analysis to predict a player's readiness to play. Our scope is the extent of the commercially deployed PMSys platform.

- We use the dataset available from PMSys as of January 2022. This dataset comes from two different elite soccer teams from the Norwegian Toppserien and includes subjective self-reported data from female soccer players over a period of two years from 2020 to 2021. This study is solely based on the readiness parameter found in this dataset.
- The experiments are carried out with one player from each team. As not all players are reflected in the data for both years, evaluating the statistics of all players for both years is impossible.
- We use LSTM for time series predictions. We elaborate on our choice of this specific ML method in Section 2.5.4.
- In our experiments we focus on predicting the next day or the next week (a total of 7 days), using data from the last one/two/three weeks, based on models trained on multiple months of data. We do not consider other prediction scenarios, as these are the most relevant in the practical context of an elite soccer team participating in a national league.

- When it comes to time series analysis, a well-known problem is the lack of data. The dataset used in this study is no exception, with numerous major gaps during days where players have not reported data. We undertake data imputation in these cases, where the previous day's data is used to fill in the value for each missing day. However, we do not extend or augment the original dataset in any way other than this specific data imputation method, which is discussed in depth in section 5.4.
- We only use freely available open-source software in our framework, in particular the Python programming language and the Tsai [45] deep learning library.

1.4 Research methods

The field of Computer Science is still young and constantly growing, but few old principles remain up to date. Peter J. Denning et. al [1] introduced in 1989 a report to suggest a framework for new ways of thinking to solve problems in the field of Computer Science and all fields of engineering. The framework by ACM is divided into three paradigms: theory, abstraction, and design.

This thesis is built on a foundation of both abstraction and design principles. This is represented by the notion that previous player and team values may be used to predict future player readiness to play values. The experiments and implementations are carried out and analyzed. The third paradigm, design, is also supported since the system's performance must be of a certain quality and follow given specifications, requiring development and testing in order to achieve acceptable accuracy.

1.5 Ethical considerations

In the realm of sports, the use of tracking devices and handling of sensitive information has become more common. With this comes additional responsibility. The protection of the data collected by the athletes has been taken very seriously. The data obtained is collected with consent from each athlete. In addition each athlete is fully aware of what data is being collected and how it is used.

The data in this research is exempt from further user consent because it has been certified by the Norwegian Privacy Data Protection Authority and is fully anonymous. All metadata is removed, and file names are generated at random.

On our behalf, we made sure that the data was fully anonymous during each process in the pipeline. This means that all the endpoints operate with newly created user id's that cannot be back translated to the athlete. The logic for mapping the user names and information to a unique user is separated from the rest of the application and is further explained in section 2.3.2. Privacy issues concerning cloud storage and usage of PMSys is further discussed in section 5.5.

1.6 Main contributions

The most significant contribution of this research is to join the ranks of the few existing works on female soccer athletes. According to a study conducted by Kirkendall in 2020 [19] only 25% of the research papers in soccer include women. Even though the interest and attendance numbers for female soccer is increasing, the numbers of research covering female soccer players is still not at a desired amount. This study is also unique among publications referring to the PMSys

system in that it consists of data from female soccer players, as opposed to prior studies that used data from PmSys. Furthermore, this research follows the players for a longer period of time than the previous studies. In this case, data is tracked over two years, which provides a broader picture and more exact numbers than short time series.

Our main goal is to investigate if it is possible to predict readiness for elite female soccer athletes using ML on data collected using an athlete monitoring system. We design and implement a software framework for running time series predictions which can be configured extensively, in order to investigate this question. Using our framework, we conduct a research study to answer four sub-questions centered around to this main question, which are specified in section 1.2. There are four key takeaways from this study:

- We showed that it is possible to make relatively accurate predictions about elite soccer player readiness using Machine Learning (ML) algorithms such as LSTM.
- We found out that training models on an entire team have advantages over using data from a single player. This is a first step toward determining the system's generalizability so that it may be used in other sports.
- We found out that the best prediction accuracy is achieved when the model is trained on one week, and predicts the next day. What this says about system implementation is that, every day prediction can be run as a cron job to give the coaches fresh statistics everyday.
- We saw that in order to not overfit, proper hyperparameters need to be selected. There is a trade-off between specific (custom-tailored) models, for instance for practical use by a team, and generalizable (e.g., use case by researchers, sports news, overall league management).

We provide a discussion of our results, and provide insights on how our findings can be used within a practical context (which might be interesting for soccer teams and developers worldwide), as well as insights on a number of research challenges (which might be relevant to many scientific communities including medical professionals, data scientists and sports scientists).

We believe that a system such as PMSys, coupled with our software framework for predictions of selected metrics, can help avoid injuries, and increase athlete performance. When training in accordance to the needs of the body, the body will then have less injuries. This will lead to better performance in the teams as each injury is costly and can contribute to cost savings and a longer carrier for the players. Since the players can train at their best and hence perform better, this can serve to increase the visibility of female soccer players, so they can have equal opportunities as the male soccer players.

However, the female body is extremely complicated, therefore more study is needed to acquire a better insight and knowledge of how to get it to work at its optimum. We refer readers to Section 6.2 for future work aspects such as this, which have been out of the scope of this thesis.

1.7 Outline

- **Chapter 2 Background and Related Work** introduces the Female Football Centre (FFRC) and how they contribute to the female elite soccer players. In order to give an insight into the dataset used in the implementation, the functionalities of the player monitoring system PMSys is presented.

Furthermore, Time Series Analysis and ML/ AI has been included in order to offer a better grasp of the framework's core concepts.

- **Chapter 3 Design and Implementation** explains the implementation of the framework developed to predict readiness to play using two years of data from two different Norwegian elite Female soccer Clubs. Missing data problems are addressed and the technologies and algorithms used are thoroughly discussed.
- **Chapter 4 Analysis and Evaluation** presents the analysis and results of the conducted experiments as well as the contextual relevance.
- **Chapter 5 Discussion** discusses the results, including use cases as well as ethical concerns with the data collection, the framework and limitations.
- **Chapter 6 Conclusion** concludes the thesis by the results from previous chapters and the outcome of the thesis with suggestions for further work.

Chapter 2

Background and Related Work

2.1 Female soccer Research Centre (FFRC)

This thesis was conducted in corporation with Female soccer Research Centre (FFRC). FFRC [9] is a research centre dedicated to women's soccer located in the heart of Tromsø in Norway. The main goal of the centre is to gain new and fundamental insights into what affects the performance and overall health of female elite soccer players. A general objective is to devise novel methodologies for epidemiological research that might impact research fields in both sports and medicine. In particular, FFRC aim to develop a non-invasive, privacy-preserving technology that allows to continuously quantify and monitor athlete behavior where analytic insights are derived from different perspectives such as biomechanics, sports-specific science, medicine, coaches and athletes.

In the current gold standards for epidemiology, observational prospective cohort studies include that cohort subjects are followed in detail over a longer period of time. This is an error-prone and tedious task that has for a long time been carried out using pen and paper, and later doing a manual, tedious analysis. Making this entire process easier is the main responsibility of the researchers from the Department of Holistic Systems at SimulaMet.

In cooperation with UiT and Forzasys AS, SimulaMet has earlier developed and used an automatic performance monitoring system for athletes used by both national and elite series soccer teams. The goal is to quantify and develop accurate analysis technologies that enable a personalized assessment and performance development of elite athletes.

The automatic performance monitoring system collects athletes' subjective parameters, like training load, wellness, injury, and illness, using a small questionnaire-app running on their mobile phones. The data is transferred to a cloud-based backend system, with an integrated automated testing and deployment pipeline. Then, from a trainer-portal, the data can be automatically visualized for both individual players and team overviews.

In FFRC, the objective is to extend the system further to include female-specific parameters and introduce more automatic analysis using, for example, machine learning. FFRC will host and build the system for all seven teams involved in the project. They also launch automatic studies that may be able to detect future overuse injuries or assist in determining the best development process for a player or a team.

2.2 Athlete Monitoring and Digital Health

2.2.1 Injury Tracking and Prevention

Injury tracking refers to the process of keeping track of the injuries that athletes have experienced. This is so that after the player's injury and rehabilitation, the load can be adjusted. If this is not done, the athlete may become injured again after a short period of time, resulting in the player being unavailable for significant parts of the season. Because athletes are constructed differently, each player's recuperation period varies. As a result, subjective reports of injuries experienced by the player are critical. In a study conducted by Eliakim et al., 2020 [7], they discovered that 136 days of injury may cost an English Premier League team one league point. The consequences were more severe with 271 days of injury in the team, including a drop in league position. soccer clubs have a vested financial interest in investing in injury prevention and rehabilitation.

Eliakim et al., 2020 [7] also concluded that injury prevention could be quantified not just in terms of rehabilitation, but also how much money they put into their sports science department, which includes professionals in sports medicine and nutrition. It demonstrates how invested a soccer club is in injury prevention when combined with investment in technology that analyzes multiple variables such as GPS, camera-based tracking, and biomechanical screening, among others. The cost of closely monitoring the players; expenditures in technology and expertise in numerous sectors pay off in the long term when contrasted to the costs of injuries.

2.3 PMSys Framework

PMSys is an athlete performance monitoring system [34, 33, 17] developed by Forzasys [11]. Previously, training performance and wellness variables were manually recorded using pen and paper. PMSys was created to replace this time-consuming, manual data collection with something more user-friendly and available. Through a subjective questionnaire answered through a mobile phone app, the system collects and analyzes athletes' training load (RPE - rate of perceived exertion), general wellness, and injuries. PMSys is a smartphone application for both iOS and Android that allows athletes to effortlessly record their statistics and track their own progress over time. There is also a trainer web page where coaches can visualize team and/or individual player statistics, and the system reports basic patterns based on the information. Reminders for reporting can be delivered via push notifications in the app and coaches can specify the expected training load.

Individuals may track their own progress, and team coaches can measure the workload and performance of their teams. PMSys is the result of a collaboration between Simula Research Laboratory and the University of Tromsø, as well as ForzaSys. The Norwegian School of Sport Sciences, the Norwegian national soccer team, and a number of teams in The Elite Series have also provided important advice and comments. ForzaSys is now in charge of the system's hosting and development.

Training performance and wellness metrics were previously recorded manually with pen and paper. The intention with PMSys was to replace this cumbersome, manual data collection with something more user friendly.

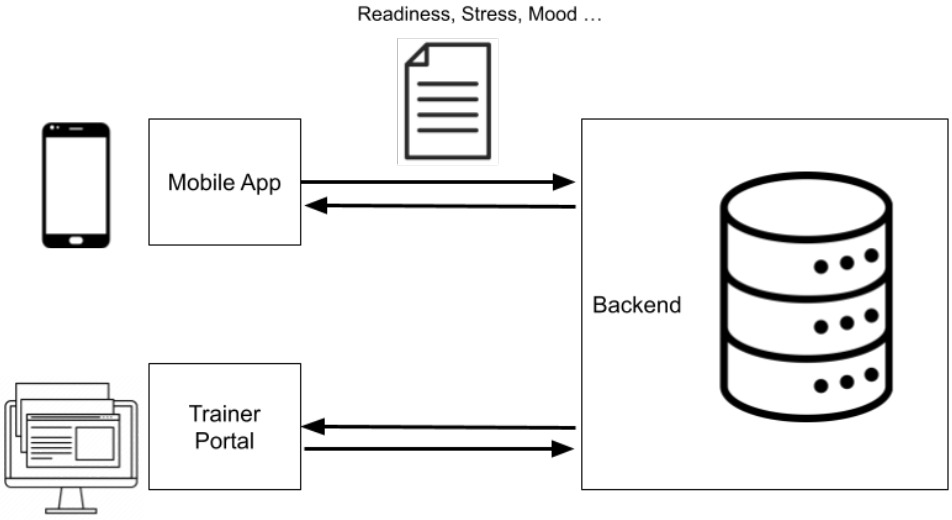


Figure 2.1: Current PMSys data collection framework.

2.3.1 PMSys Frontend

As previously mentioned, the architecture of PMSys consists of multiple elements. The main components that make up the frontend of PMSys is the Mobile application for data collection and the trainer portal. The mobile application has a user-friendly interface that has been properly tested and been improved by the feedback from the users. As a result the process of tracking wellness parameters and current injuries require very few clicks from the user. Figure 2.2 shows the user friendly interface with restricted amount of information in order to keep the reporting an easy task for the user.

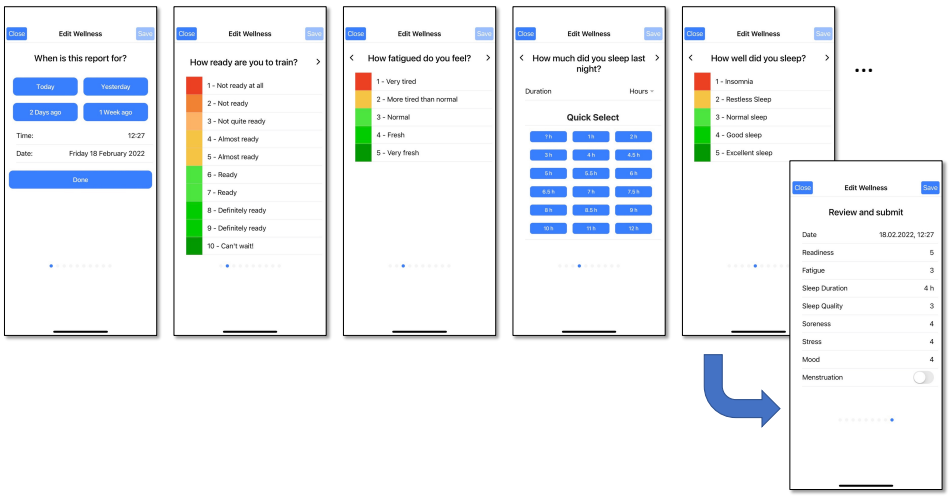


Figure 2.2: Reporting of wellness in the mobile application.

After the user enters the data, the frontend sends it to the backend for further persistence. Once the information is saved in the database, the trainer is able to see the reported values for a given player. Figure 2.3 displays how the user interface looks from a coach's perspective. We can see that this player has reported

a readiness score of 8, which is in the upper range of the readiness scale. The injury report indicates that the player has been injured in both feet around the shin/calf area. The training portal gives the trainer a complete overview, so that the training can be adjusted accordingly. Figure 2.4 displays a plot showing the training load for a player in the period of 21.07.2020-20.08.2020. This is one of the many plots that are available in the trainer portal.

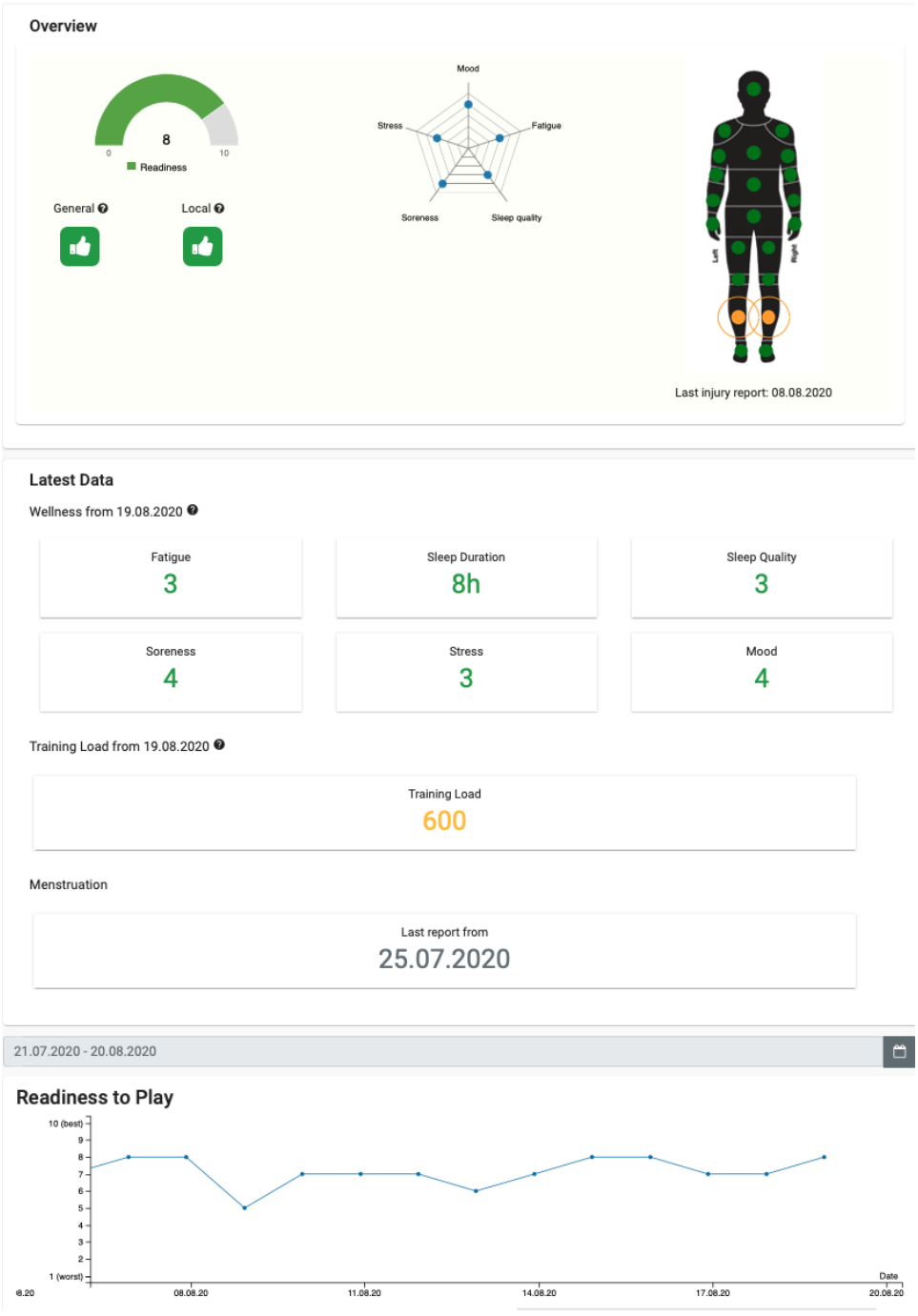


Figure 2.3: Trainer portal with wellness parameters for a player.

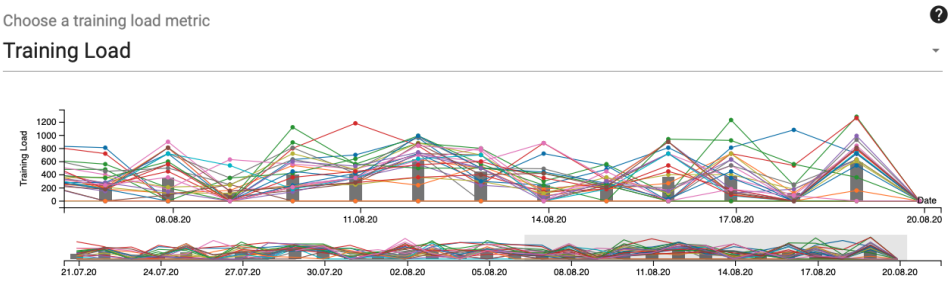


Figure 2.4: Training load from trainer portal.

2.3.2 PMSys Backend

The architectural style for the PMSys backend is cloud-based with microservices that run on the Amazon(AWS) public cloud [17]. Once the data is sent from the real-time questionnaire to the backend, a DSU receives the data and sends it further for persistence. With the exception of a cloud-hosted PostgreSQL database for permanent state, the DSU is kept functionally minimal, reliable, and independent from other system components. This is done with criticallity in mind. DSU is a collection of servers, and player reports can be dispersed across them based on the amount of isolation or replication necessary [49]. To protect the participants' identities, everyone is given a unique identifier, which allows the athlete data to be stored anonymously in the system.

Players with iOS and Android phones can use the app since it is platform independent and runs on both platforms. An example of the questionnaire is displayed in figure 2.2. The program is divided into several forms and is designed to have as few clicks as possible to make athlete diary input a quick and easy task. Players are notified via push notifications on their mobile phones, so they do not have to keep mental notes on when to register the data.

PSU is a component that manages rules such as coach/trainer access, aggregation functions, and time-restricted access. The data owner is in charge of the PSU. The DSU has limited data, and accessing it requires gathering profile information from the PSU in order to create an identity. This will enable regulations to allow for the real-time transmission of pseudonymous player data to aggregation functions and deep learning. [49].

2.4 Time Series Analysis

A collection of observations taken sequentially in time is defined as a time series. The times at which the observations were taken can be regularly or irregularly spaced. The time measurement is what distinguishes a time series from a regular data sample. Time series are frequently used to predict future occurrences and analytics. There are no limitations to the features that can be predicted. Economic forecasting, sales forecasting and stock market analysis are very common. This thesis will concentrate on predicting soccer players' readiness to play for two different Norwegian soccer clubs. The main focus will be a univariate time series. A univariate time series focuses on a single given feature. In our case it is readiness to play.

2.5 Machine Learning

ML is a field of computer science which uses computational algorithms to analyze, learn from historical data, and then simulate human intelligence by learning from the environment [48]. A machine learning algorithm's fundamental objective is to learn from a current context and be able to generalize to apply and obtain the same results into unseen task [48, 4]. Machine learning techniques have helped pattern recognition, computer vision, spacecraft engineering, economics, entertainment, and computational biology, as well as biological and medical applications. [48]. The upcoming sections are meant to give a brief introduction to core ML concepts in order to fully comprehend the reasoning behind the decisions around the algorithm used in the implementation.

2.5.1 Supervised Learning vs Unsupervised learning

ML techniques are usually divided into two approaches: supervised- or unsupervised learning. Supervised learning uses data points where the relationship between the data entry and the label is clear. This approach is suitable for classification and regression problems. Unsupervised learning has the ability to learn patterns in the data without further assistance on how it should be executed [14]. Typical areas of use for unsupervised learning is clustering, reduction of dimension and insight gathering in prior to designing a classifier [41].

2.5.2 Regression

Regression is type of supervised learning method that has the aim to predict a numerical value given an input [14]. This is done by understanding the relationship between dependent and independent variables [41]. Regression is helpful in our case, as we aim to predict a numerical value of readiness and observe how it reaches the actual values.

2.5.3 Generalization and Model Optimization

As earlier mentioned, the main objective in ML is to train the model to recognize general patterns on unseen data. Generalization refers to the ability a model has to adapt to new and unseen data points [14]. The training error is computed based on the training set during machine learning training, and the goal is to decrease the training error to a reasonably low value. Additionally, during the test phase, generalization error is calculated by computing test error using the test set. Underfitting occurs when the model is unable to obtain desirable low training error. Overfitting occurs when the gap between the training and testing error is too big. To tackle the issue of over- and underfitting there are some model optimization that can be done. A models complexity is often determined by a number of hyperparameters [14]. These parameters can be modified in order to optimize the model. Figure 2.5 visualizes the phenomena of underfitting and overfitting during each training iteration determined by epoch.

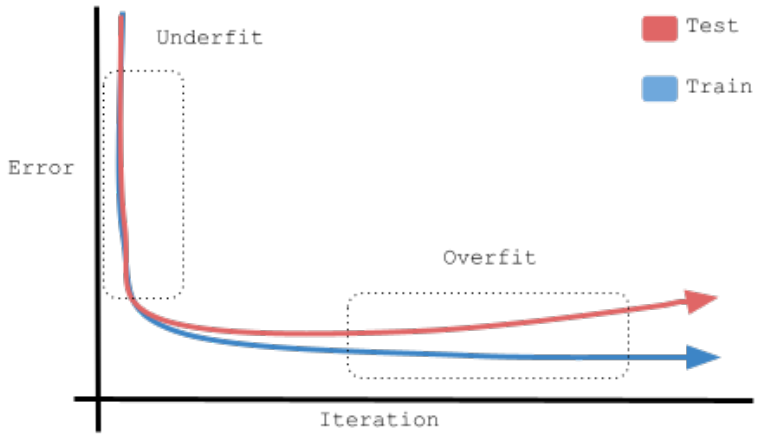


Figure 2.5: An overview of the relationship between the size of the error when training vs testing. Highlighting when the model is underfit and overfit.

2.5.4 RNN and LSTM

Deep learning is a subfield of ML concerned with algorithms that mimic the functionality of a human brain [47]. By imitating the biological brain structure, the neural networks consist of a layered network of nodes of specific structures named neurons [42]. Recurrent Neural Network (RNN) is a type of neural network that uses the hidden state to function as an internal memory (see Figure 2.6 for details). RNN is suitable for processing time-series data since it also considers its context during the training.

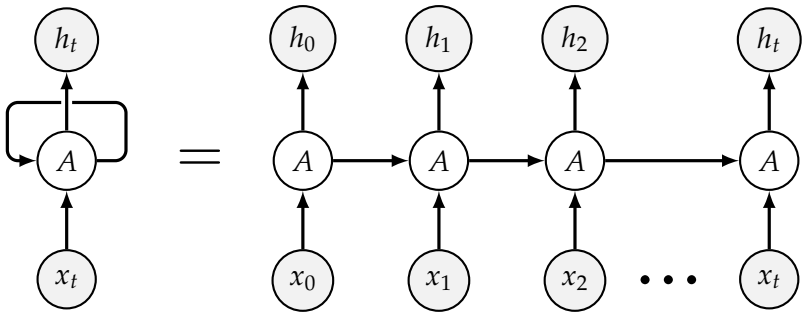


Figure 2.6: RNN architecture where x_t represents the input, A represents the hidden layer that contain previous state and finally h_t is the output. Figure from [39]

Long Short-Term Memory (LSTM) is a recurrent RNN architecture that outperforms regular RNNs on a variety of temporal processing tasks [5]. Unlike RNN, LSTM manages to solve the problem with short memory that RNN suffers from. LSTM has the ability to manage, forget and ignore data points based on a probabilistic model by using a sequences of gates [24]. Each of those gates has its own RNN (see figure 2.7. The ability of Long Short-Term Memory (LSTM) [16] to predict future values based on prior sequential data is one of the reasons it is a suitable method for multi-step univariate time series prediction. LSTM is capable

of solving various time series problems that feedforward networks with fixed size time windows cannot. Due to its ability to learn long-term correlations in a sequence, LSTM networks eliminate the need for a pre-specified time window and can adequately simulate large multivariate sequences [28]. Furthermore, LSTM gives flexibility to the framework, allowing it to be modified if it becomes essential to integrate multi-variable variables to forecast future readiness to play. Since PMSys keeps track of several wellness parameters, which are given in table 3.1, it will be possible to use those and evaluate if the accuracy of the predictions improves.

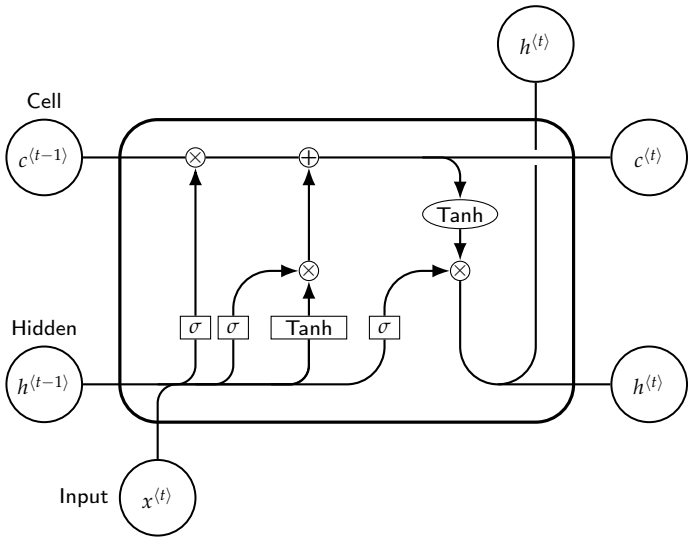


Figure 2.7: LSTM architecture overview where the cell state c^t that it can choose to read/write from it or reset it. (Modified from [23])

LSTM works better than traditional statistical approaches as ARIMA [49] and shown by a study conducted by Niamini et. al [40] LSTM had a 88.07% reduction in the calculated MSE compared to ARIMA. Wiik et al. [49] confirmed that LSTM outperformed ARIMA when predicting readiness to play. In addition, Ma [26] came to the same conclusion when predicting stock prices. This is why we have chosen to focus on LSTM and disregard traditional statistical approaches in this work. Furthermore, we did not observe periodic behaviour in readiness data, therefore such simplistic approaches might not be adequate for modeling.

As mentioned earlier in chapter 2, the scope of the thesis was a univariate time series analysis. Our data analysis framework can be suitable for teams who prefer a lightweight implementation, which only rely on a single parameter.

2.6 Related work

Several experiments have been carried out to see if machine learning can be used to forecast future values for soccer players' readiness to play in collaboration with Forzasys. Wiik et al.[49] conducted a study with the purpose of reducing sports injuries and predicting readiness to play. Based on a dataset from two male high division soccer teams in Norway, they demonstrated the value of utilizing a LSTM RNN to predict reported training load. They were able to train the model to predict positive peaks and negative peaks. Positive peaks are categorized as values above 8 and negative peaks are categorized as values below 3. Both of the datasets did not have values for all days. As a result, they had to account for the

issue of missing data. The missing data were not replaced or deleted to provide a realistic use scenario. This was done to provide a more realistic use case, as data gaps would always exist due to vacations, injury time, and other factors. The data for the first time was from January 2017 until late August 2017 and contained readiness to play for 19 players. For the second team the dataset consisted of data from February 2018 to mid June 2018 for 22 players. It was 6000 entries in total.

To keep the simplicity of reproducing and interpreting the results, the model was kept small. This allowed to explore the underlying possibilities in the data.

The values for the hyper parameters were respectively 36 and 30 with a batch size of 4. To assess and confirm the data, two distinct methodologies were used. The initial strategy was to train the entire team and then predict when the player would be ready to play. The second strategy involves training the model on the player who will be predicted. When the entire team was used to train the model, the predictions were more accurate, and the graphs closely tracked the peaks. Wiik et al.[49] attempted another experiment with traditional machine-learning methods like linear regression and Random Forest, but the results were not significantly improved.

Another research conducted in a collaboration with Forzasys is a study done by Sierhei Kulakou [20]. In this study, 13 models from the machine learning library, Tsai, were used to to predict readiness to play using data from PMSys. LSTMPlus is one of the models which is employed in the present research. The data consisted of 34 unique players over a period of 7 months.

Kulakou had as a main goal to predict the past 10 days using the training models. To test the trained models, the parameters tweaked was the type of the input data (univariate vs. multivariate), sliding window size, data with and without gaps filled with 0 value, number of epochs and the amount of input data (training on single player vs. on the entire team concatenated). The number of epochs was set to 200 and batch size to 128. The sliding window was tested with a broader range from 3, 5, 7, 14, 21, 28, 35, 42 and 49.

The findings of this research revealed that the models with the lowest MSE differed in terms of which data was used to train them and which data was used to test them. LSTMPlus was one of the models who was able to successfully predict negative and positive peaks. For uni-variate series the model performed best with epoch value of 100. A value of 200 resulted in over-fitting. For multivariate series, the number of epochs did not have to pass 20 before the problem of over-fitting arise. He discovered that a sliding window of 3 produced the best results. The lowest MSE reported was 1.325 for multivariate time series and was returned by LSTMPlus. For the univariate series the lowest MSE was 1.381 performed by regular LSTM. The results showed that training on the entire team and leaving one player out returned the best results. Both negative and positive peaks matched the actual result more closely when training on the entire team. The findings of Kulakou's study are equivalent to those of Wiik et al.

Another research done by Johansen et al.[17] demonstrated the impact of employing current technologies in sports to detect injuries and train optimally. In this study they have seen the advantages of incorporating technology into elite athlete performance. Their decade of expertise has culminated in a smartphone-based application with a backend system for cutting-edge athlete monitoring. A cooperation between computer scientists, sport scientists, and medical professionals has helped to discover gaps that technology can address, allowing athletes to progress in the proper direction. PMSys was well-received by both athletes and staff, making it simpler to advocate for earlier bedtimes and modified training days.

2.7 Summary

In this chapter, we have provided an overview of the partner of this thesis, FFRC, the importance of athlete monitoring and injury tracking and the athlete performance monitoring system, PMSys, that has provided us with the collected data. In addition, we have taken a look at other similar projects to see how ours differs. Necessary background information on concepts including deep learning, RNN, a detailed justification on the choice of algorithm and provided extensive context for the project have been presented.

Chapter 3

Design and Implementation

The value of injury tracking and prevention was explained in Chapter 2, including the essential functionality in the PMSys system, fundamental principles in Time Series Analysis and Machine Learning and reasoning for why LSTM. Finally, a sample of related work is presented, together with a discussion of the similarities and differences.

This chapter covers the technical implementation of the framework, as well as explanations of technology choices and where this framework fits in PMSys.

PMSys [34, 33] is today used by several elite soccer teams collecting and visualising wellness and training load. However, such data is typically used to plan future training, thus it would be of great value if the system also could make predictions for future readiness and identify unwanted trends for example typically resulting in overuse injuries. Machine learning has the potential to be a promising solution for such prediction [49, 17]. In this chapter, we will provide a detailed explanation of the technical requirements for developing a system that uses LSTM and a state-of-the-art Deep Learning library to predict future performance in terms of readiness to play. This will make it easier for soccer teams and other sports in different disciplines to tailor the training load to fit the athletes' performance and prevent injuries based on self-reported data. In order to gain a thorough comprehension of the system and eliminate the possibility of ambiguity, a few key principles must be understood. These principles will be presented first in this chapter.

3.1 Overview and Terminology

As numerous of the key concepts may have multiple meanings in different settings, a list containing explanations of the most fundamental concepts has been created.

Terminology:

- **Input window** is the historical data used to conduct a single prediction. It is also known as the look back window for the prediction. An input window of 7 indicates that 7 days of the dataset is being used.
- **Output window** is the future time steps beyond the provided data as input. These future time steps are also commonly known as the horizon. An output window of 4 will result in the framework predicting the readiness to play for the 4 upcoming days.
- **Sliding window** is the use of prior time steps to predict the upcoming step.

- **Batch size** is the number of total training samples provided in a single epoch.
- **Run** or an iteration is the number of times a batch of data is passed through the model.
- **Epoch** is a single forward and backward pass of the entire dataset through the neural network.
- **Athlete** is used to refer to one of the teams' soccer players. This is equivalent to an individual player.
- **Trainer** refers to a coach of a team.
- **Match** is another word for a game with a typical duration of 90 minutes.
- **Season** in this context is a period of time during which the soccer team competes in matches. Typically, the season runs from April until November.
- **Training session** is in this circumstance a practice session. This is usually done as a team, but it can also be done individually.

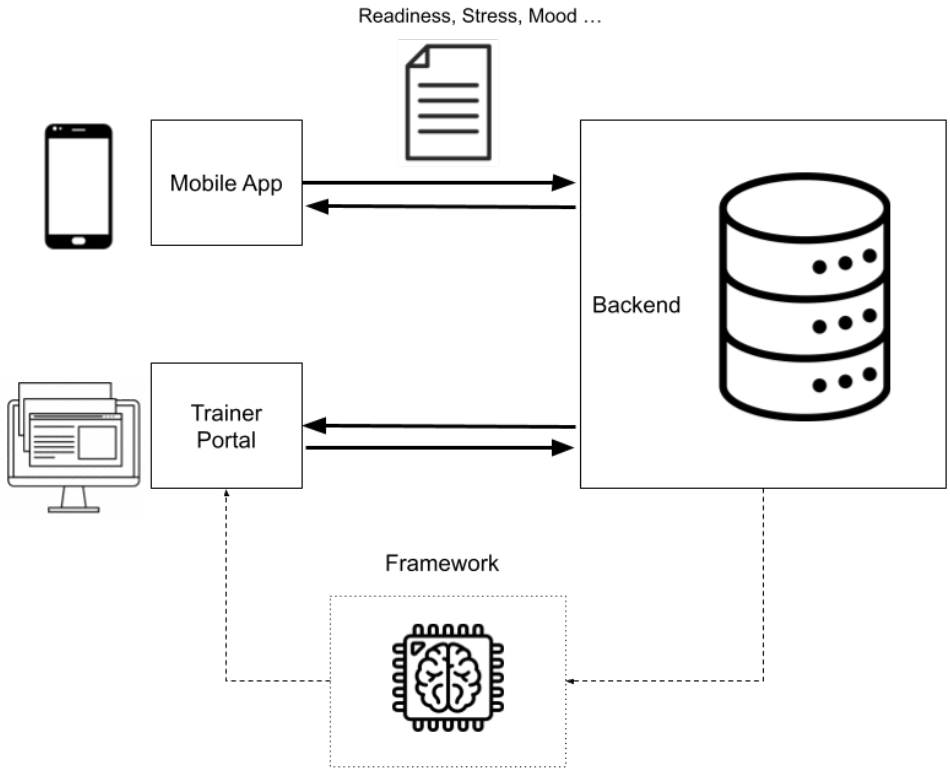


Figure 3.1: Overview of where the framework fits in the architecture.

Overview:

As earlier mentioned in section 2.3.2 the data from backend is unprocessed and the trainers can view simple visualization of team and individual player statistics in the trainer portal. This is visualized in figure 3.1. The new framework for predicting athletes' future performance can be added without having to rewrite the existing system. The dotted lines show that this is not a new dependency that has been added. It is rather an optional extension that they can choose

whether or not to utilize. By doing so, we achieve higher flexibility by avoiding the introduction of dependencies that can make the system more difficult to maintain and upgrade in the future.

3.2 Dataset

The datasets provided by PMSys consists of data from two different female elite soccer teams in the Norwegian league. Per year, each team has their own dataset. This is for the years 2020 and 2021, and it covers the months of January 1 to December 31. The number of soccer players in each dataset varies, as does the amount of data records for each day. To make it simpler to keep track of the teams, they have been named team A and team B. In team A, 15 players are present in 2020, whereas 23 players are present in 2021. In 2020, team B has 24 soccer players and in 2021, team B has 28 soccer players. For team A, the percentage of days with missing data is 61.57 percent in 2021 and 32.65 percent in 2020. Whereas for team B, it is 42 percent in 2021 and 54.07 percent in 2020. There are a lot of gaps in the datasets, which might have a big influence on the final results. Backward fill is the strategy used in this thesis to solve this problem. This signifies that for the following day's missing value, the prior value is utilized. For example, if Monday has an 8 for readiness to play and Tuesday is lacking a value for a specific player, the dataset will be changed such that Tuesday reflects Monday's value of 8.

Table 3.1 presents an overview of the number of players on each team for the year 2021 and 2020, as well as the percentage of missing data for each year. This percentage is derived by the sum of all the players' readiness records for an entire year. This is then divided by the number of days in the year, and multiplied by the number of players.

Team	**Year**	**No. of players**	**Missing data**
A	2021	23	61.57%
B	2021	28	42.00%
A	2020	15	32.65%
B	2020	24	54.07%

Table 3.1: For the years 2021 and 2020, the table below shows the distribution of player numbers and missing data for teams A and B.

Table 3.2 presents a list of parameters collected by the framework.

Category	Field	Description	Data Type	Range
Metrics	Fatigue	The current fatigue level of the player	Numeric	1-5
	Mood	The current mood of the player	Numeric	1-5
	Readiness to play	The athlete's readiness for a training session or a game	Numeric	1-10
	Stress	The current stress level of the player	Numeric	1-5
	Sleep quality	The quality of the sleep	Numeric	1-5
	Sleep duration	The duration of the sleep	Numeric	0-12
	Soreness	The level of soreness	Numeric	1-5

Table 3.2: Several wellness metrics can be found in the PMSys dataset. These include stress, sleep quality, sleep duration, mood, and readiness to play, all of which are essential in this research.

The dataset from PMSys contains several wellness parameters. Stress, sleep quality, sleep duration, mood, and finally, readiness to play are the most important factors in this research. These parameters are also affected by the problem of missing data. It is conceivable that performing a multi-variable prediction could lead to large uncertainties in the predictions due to the nature of compounding uncertainty for each filled value. It is possible that a combination of these may lead to higher precision of the prediction. If there is a correlation between the variables, it's likely that the missing data problem will be mitigated if you have real data for multiple parameters. This is given that the amount of data affects the model.

A little pre-processing was required in order to be able to work with the given data. The readiness to play values required to be converted from integers to floating numbers. This is because Tsai wants input data such as floating numbers to create splits. In section 3.4, this is explained in further depth. There were no further changes to the dataset besides the conversion from integers to floating numbers and the filling in of the gaps by backward propagation for missing values.

The number of entries in a dataset is an important component of high-quality research since it determines the statistical power of a test [27], and thus it important to be aware of the impact the size of the dataset can have. This is especially the case for team A year 2020 where the number of players is all time low with the value of 15 players. This might have significance for Team A's testing results in the year 2020.

Because the datasets include thousands of rows of data, it is difficult to see how the data is dispersed at first look. As a result, a comparison of both teams for both years has been made, along with the related mean and standard deviation (std). The standard deviation measures the variability in the data collection on average. It indicates how far each value deviates from the mean on average. Data are grouped around the mean when the standard deviation is low, while data are more spread out when the standard deviation is high.

As can be seen in table 3.3, there are few players on team A in 2020 who have a high standard deviation. This indicates that the values are not subject to a lot of fluctuation. Table 3.3 shows mean and standard deviation for team A's readiness to play per player for the year 2020.

Player	Mean	Min	Max	STD
1	7.58	5.0	9.0	0.78
2	7.70	1.0	9.0	0.92
3	6.22	1.0	8.0	3.03
4	7.48	2.0	9.0	0.81
5	7.16	1.0	8.0	0.76
6	8.66	5.0	9.0	0.65
7	6.20	3.0	8.0	0.71
8	5.77	5.0	8.0	0.71
9	6.15	1.0	10.0	1.29
10	5.95	4.0	6.0	0.29
11	5.69	2.0	6.0	0.82
12	6.74	2.0	10.0	1.58
13	6.28	1.0	10.0	1.35
14	6.33	6.0	10.0	0.93
15	6.59	3.0	9.0	1.01
16	6.23	1.0	7.0	1.09
17	6.99	5.0	8.0	0.73
18	7.98	4.0	10.0	0.88
19	6.99	4.0	8.0	0.90
20	6.26	4.0	7.0	0.80
21	8.16	3.0	10.0	1.49
22	6.93	5.0	9.0	0.34
23	6.93	3.0	9.0	0.47

Table 3.3: An overview of the mean, min, max and std of readiness for team A year 2020.

Table 3.4 presents the mean and standard deviation for team A's readiness to play in the year 2021. It shows that there are more players with a higher standard. As a result, a higher spread in Team A's values might be expected in 2021.

Table 3.5 reveals the mean and standard deviation for team B's readiness to play per player for the year 2020. In terms of the standard for the year 2020, Team B resembles Team A in certain ways. There are few athletes who have high fluctuation in their values.

Player	Mean	Min	Max	STD
1	7.33	2.0	10.0	1.13
2	7.00	7.0	7.0	0.00
3	8.22	5.0	10.0	0.76
4	6.81	5.0	9.0	0.55
5	7.69	1.0	10.0	0.80
6	8.35	5.0	10.0	1.02
7	7.05	3.0	9.0	1.00
8	5.78	1.0	8.0	0.82
9	8.97	6.0	10.0	0.52
10	6.71	4.0	10.0	1.08
11	5.42	2.0	7.0	0.93
12	8.06	1.0	10.0	2.02
13	6.81	3.0	8.0	0.68
14	9.98	9.0	10.0	0.16
15	4.85	1.0	9.0	2.56

Table 3.4: An overview of the mean and std of readiness for team A year 2021.

Player	Mean	Min	Max	STD
1	7.49	2.0	10.0	1.68
2	6.07	1.0	8.0	0.82
3	6.33	3.0	10.0	1.03
4	6.22	3.0	9.0	0.80
5	5.46	3.0	8.0	0.94
6	6.83	4.0	9.0	1.03
7	6.72	3.0	10.0	1.21
8	7.14	4.0	8.0	0.68
9	8.26	5.0	10.0	1.25
10	6.07	1.0	10.0	1.80
11	6.53	2.0	10.0	2.35
12	6.48	4.0	9.0	0.97
13	6.59	3.0	9.0	1.17
14	6.32	1.0	9.0	1.34
15	5.10	1.0	8.0	0.83
16	6.80	3.0	8.0	1.05
17	7.39	4.0	9.0	0.80
18	6.99	3.0	10.0	0.73
19	8.37	5.0	10.0	0.73
20	6.30	3.0	8.0	0.80
21	8.99	4.0	10.0	1.62
22	8.22	5.0	10.0	1.02
23	6.78	2.0	8.0	1.74
24	7.78	3.0	10.0	0.74

Table 3.5: An overview of the mean and std of readiness for team B year 2020.

There are more players with a standard deviation greater than one on Team B in 2021, resulting in more fluctuations in the values. Table 3.6 shows the mean and standard deviation of team B's readiness to play per player in 2021.

Player	Mean	Min	Max	STD
1	6.25	3.0	10.0	1.50
2	6.36	2.0	9.0	0.90
3	6.01	5.0	7.0	0.10
4	3.73	1.0	8.0	1.68
5	6.20	3.0	8.0	0.68
6	6.75	2.0	9.0	1.35
7	7.32	1.0	10.0	1.26
8	6.78	5.0	9.0	0.69
9	7.84	3.0	9.0	0.94
10	6.72	1.0	10.0	1.39
11	6.93	2.0	10.0	1.17
12	7.27	1.0	10.0	1.52
13	6.15	1.0	8.0	1.18
14	4.96	1.0	8.0	1.00
15	6.62	1.0	10.0	1.06
16	7.19	3.0	8.0	0.66
17	6.86	1.0	9.0	1.25
18	7.89	6.0	9.0	0.35
19	6.33	3.0	10.0	1.04
20	6.67	1.0	10.0	2.15
21	6.89	1.0	9.0	1.08
22	5.97	3.0	8.0	1.11
23	7.65	4.0	10.0	0.85
24	8.67	5.0	10.0	0.72
25	6.28	5.0	7.0	0.53
26	7.15	2.0	9.0	1.26
27	8.92	2.0	10.0	1.49
28	8.82	1.0	10.0	1.79

Table 3.6: An overview of the mean and std of readiness for team B year 2021.

The value of the tables which displays the mean can be of greater benefit later in the research when the need to choose players for testing emerges. This will be covered in further depth in section 3.4.1.

3.3 Design Choices

The main goal with the research is to investigate how we can predict readiness for elite soccer athletes based on LSTM and historical data. Furthermore, it is important to determine whether individual or team data is the most reliable way to estimate a player's readiness. The implementation should support different input and output window, as well as have the hyper-parameters configurable. To

avoid rewriting the solution and avoiding adding any unnecessary complexity and potential shortcomings, it is of high importance to map the technical and practical requirement of the system. Another benefit of mapping the need initially is that the options can then be limited down to fewer and clearer options. With the large selection of deep learning libraries, it is easy to get caught up in the excitement of a new great library that either lacks key functionality or proves to be overkill down the road.

To help narrow down the choices of algorithms and the scope of a time series prediction problem, Brownlee [5] has developed a framework based on the following considerations.

- **Input vs. output**. It is important to form a clear picture of what the input data and what the desired output data is. In the present thesis the clarity of the goal is not a question. The model is going to consume readiness data as input and return predicted readiness to play as output. In certain cases, it may not be as obvious and may require careful consideration.
- **Endogenous vs. Exogenous**. To better understand the relationship between the input data and the output data they can placed into two categories. Endogenous variables are dependent on the output variable and are impacted by other system variables. Exogenous variables, on the other hand, are unaffected by other model variables. The variable readiness to play is endogenous as it is dependent on the input variable.
- **Unstructured vs. Structured**. The data is called structured if it contains any time-dependent patterns, such as trends or cycles. If there is no pattern in the data, it is unstructured. By eliminating obvious structures like repeating trends and cycles, the modeling process may be simplified.
- **Regression vs. Classification**. It is a regression problem if the end goal is to predict a quantity. Otherwise, it's a classification issue if the purpose is to classify into categories. In our case, the end goal is a quantitative value indicating an athlete's readiness to play.
- **Univariate vs. Multivariate**. A univariate problem is one in which the prediction is made using only one variable as an input. A multivariate problem is one in which numerous input variables are used. It is essential to use a suitable algorithm when dealing with multivariate input, as not all support multivariate variables. The scope of this thesis is to predict readiness using a single variable, readiness.
- **Single-step vs. Multi-step**. Deciding whether it is a single step or a multi-step prediction is one of the criteria that highly impact the implementation. Single step prediction is only predicting the next time step. Multi-step prediction is predicting more than one upcoming time steps. Due to the compounding nature of the uncertainty in each step, multi-step prediction increases the complexity of the system accordingly [5].
- **Static vs. Dynamic**. Is it necessary to create a reusable static model? Is it required to use a more dynamic model, one that evolves with each prediction? In our case, having a static model is beneficial since it avoids both frequent updates and unnecessary resource consumption.
- **Contiguous vs. Discontiguous**. Are the values in the dataset contiguous, meaning that they have values for each hour, each day or each week? Or are they discontiguous with sporadically available values leaving the dataset with missing data? This is crucial to examine since the method used to handle the disjointed data in order to make it consistent might have a significant influence on the prediction.

3.3.1 Why use Python?

Python stands out as a programming language for machine learning because of its extensive set of built-in well-documented libraries and frameworks. The built-in libraries allow the users to access, handle and transform the data as desired. An example of that is Numpy [32]. It is a scientific computing package that is frequently used by both the ML community and is utilized in this thesis. The growing popularity and the strong community support is also the reason for solving problems with Python.

3.3.2 Why use the Tsai library in Python?

It is the required continuity and temporal dependence between the entries in the dataset that cause the need to use more advanced techniques such as deep learning. Tsai [45] is an open source deep learning library built on top of Pytorch [37] and fastai [8], that specializes on time series techniques such as classification, regression, forecasting, and imputation. The reason behind the choice of Tsai as the chosen deep learning library is that it is under continuous development by timeseriesAI [44] and the community is involved. The involved open-source community means it is constantly being improved, and more enthusiasts interested in machine learning and deep learning add additional features as needed.

3.3.3 Why use LSTMPlus?

A multi-step forecast is forecast with a longer time horizon than one day. There is often a need to predict further down the road than the upcoming day. A typical daily real-world usage for this is prediction of weather forecast. Predicting the upcoming day is useful but having an overview of an upcoming period makes it even more important.

Tsai does not support multi-step prediction out of the box. Any attempts to use multi-step prediction with the old fashion LSTM model results in the application crashing. That is why the developers have implemented[25] model with extended features. The LSTM model accepts inputs such as the horizon to the n days wanted to forecast. It is also possible to pass the value for the stride which allows you to decide which point in the dataset to start the prediction from. This was not used in this project.

3.3.4 Approach (Conceptualization)

In very broad terms, the intended functionality is depicted in the figure 3.2 below. The current day is also known as the starting point when T is given. The input window is depicted in blue. This is calculated using T minus the desired number of days back in time. The output window is highlighted in red. This is T plus the number of days desired to predict ahead of time.

The timeline from figure 3.2 illustrates only one run. To understand how each day and calculation of MSE takes place for each run, figure 3.3 is made to illustrate this. In this one can observe that the input window is constantly moving forward for each day that is predicted. In this example, there is an input window of 21 and the output window of 7. As this setup works, the number of MSE values needed will correspond to the equation $M = NxOutput\_window$ where M is the total number of MSE entries and N is then the number of runs in the system. 392 MSE values and an ouput window of 7 will result in 56 runs.

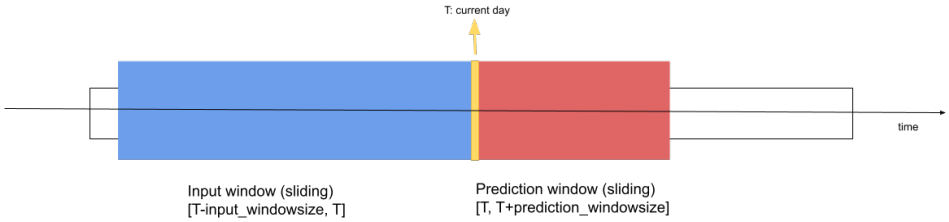


Figure 3.2: Single time series run

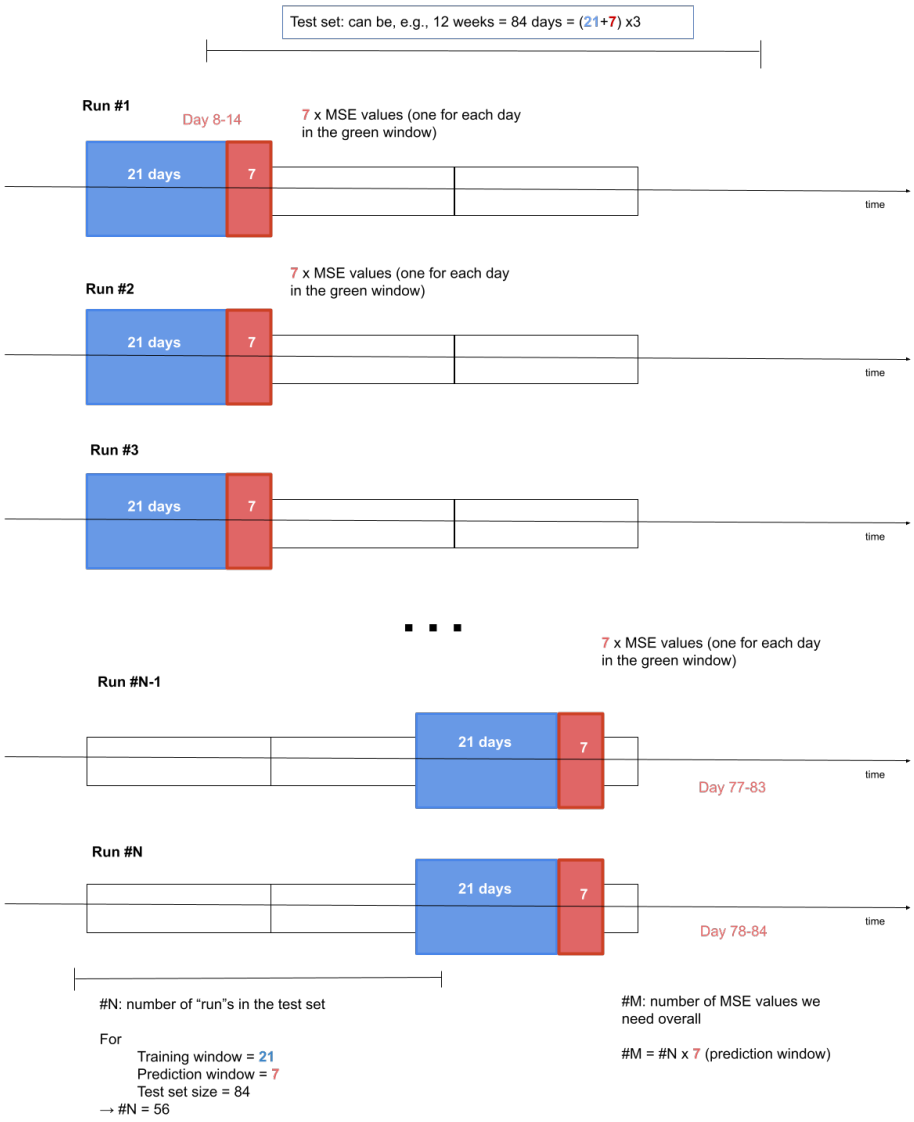


Figure 3.3: Multiple multi-step predictions

3.4 Implementation

The aim of the implementation is to develop a framework that can perform single- and multistep time series prediction given a set of pre-defined parameters. There are numerous approaches to this problem. We have chosen to use the programming language Python [36] and the deep learning library, Tsai [45]. Python [36] is a diverse programming language that helps with code structure and issue solving in a clear and concise manner. This section details the decisions that lead to the code's implementation, with each step clearly explained.

- **Code style**. The chosen style of code is imperative. This means that the code is organized in such way that there are commands that describe in which order the computer should perform those. In contrary to functional programming, where the sequence of instructions does not matter, the order in which methods are executed is of high importance.

- **Code structure**. The design principle of separation of concern is followed. This means the code is structured so that each method has one purpose. When the modularization is reached the code can be easily understood, maintained and updated without needing any major refactoring of the code. The Don't Repeat Yourself (DRY) principle is also respected by using utility functions that can be imported when needed. This is a neat way to avoid unnecessary lines of code.

- **Methods implemented**. The framework consists of a series of methods which are run in order to obtain the desired prediction. The initialization and instantiation of necessary objects such as SlidingWindow, splits and data loaders are done initially. Those objects are then passed down to the `train_and_save_model` method. This method trains the model and saves it. To test the trained model, the input- and output window, test dataset and the path to the saved model has to be provided(see listing 3.1 for details). When the model is tested with the dataset saved for testing, it returns a dictionary that contains the predicted and the actual values. Two different methods `calculate_mse` and `plot_mse` are implemented to calculate MSE and plot the MSE into a bar plot. (see figure 4.1a for example plot from this method). To plot the predicted values vs the actual values another method, `plot_preds_target`(see Figure 4.5 for example plot from this method), is implemented.

```
1    def train_and_save_model(training_validation_dataset,
     training_window, prediction_window, batch_size,
2                             epoch, n_splits, shuffle,
     model_file_name):
3    X, y = SlidingWindow(training_window, horizon=
     prediction_window)(training_validation_dataset)
4    splits = get_splits(y, n_splits=n_splits, shuffle=shuffle
     , stratify=False)
5    batch_tfms = TSStandardize()
6    tfms = [None, [TSForecasting()]]
7    dls = get_ts_dls(X, y, splits=splits, batch_tfms=
     batch_tfms, tfms=tfms,
8    bs=batch_size, arch=LSTMPlus, metrics=[mse], cbs=
     ShowGraph())
9    trained_model = train_model(dls, epoch)
```

```
10      save_model(trained_model, model_file_name)
```

Listing 3.1: Code implementation of the method that trains and saves the model

```
1   def predict_readiness(training_window, prediction_window,
     test_dataset, model_path):
2       trained_model = get_trained_model(model_path)
3       preds, targets, _ = test_model(training_window,
    prediction_window, test_dataset, trained_model)
4       calculated_mse = calculate_mse(preds, targets,
    prediction_window)
5       plot_mse(calculated_mse, training_window,
    prediction_window)
6       plot_preds_target(np.array(preds), np.array(targets))
```

Listing 3.2: Code implementation of the method that predicts readiness based on the model saved from the training method 3.1.

- **Utility methods**. There are two separate utility files with code. The file data_utils.py contains the methods that read the data-frames from excel files and the extraction of the data from the Readiness tab. The extraction method reads the file, iterates over each player and if a value is missing it will alter the value to reflect the closest historical value. This method is ran once and then persisted, so all the datasets are updated.
- **Parameters for prediction**
  - **Training dataset** is the dataset containing the values the model will utilize for training.
  - **Testing dataset** is the dataset containing the values the model will utilize for testing the trained model. This is data it has not seen before.
  - **Input window** decides how many historical days is used to train the model.
  - **Ouput window** decides how many days in the future it is desired to predict.
  - **Epoch** determine how many times the entire dataset is passed through the neural network.
  - **Batch size** decides how many training samples there will be in a single batch.
  - **N splits** specifies the number of folds. Default is 1.
  - **Shuffle** is a boolean that decides whether to shuffle the training data or not.
- **Sliding window**.To create an array of segments of a pandas dataframe based on the given criteria such as input window and the output window, the function SlidingWindow in Tsai solves this.
- **Splits**. To divide the data into a training and validation sets, the split method handle this by consuming arguments such as `valid_size` and `test_size`. It is possible to pass the `n_splits` parameter to perform k-fold cross-validation. However when passing a value greater than 1, the system crashed and the creators behind Tsai did not manage to respond in time due to an increased amount of open issues on GitHub.

- **Batch tfms and tfms** uses TSStandardize() to standardize the dataset and TSForecasting() to instanciate the with LSTMPlus as the chosen algorithm. Both are then passed to the data loaders.
- **Data loaders**. The data loaders require parameters such as the training and validation data transformed by the sliding window, the splits, `batch_tfms`, `tfms`, algorithm (LSTMPlus), metrics to calculate (MSE) and what graphs to display. The data loader is then passed to the method that trains the model and the model is then persisted after training.
- **Model size**. The average size of a model trained on a single player is 4.2MB, and trained on all players is 13.2MB.
- **Requirements**. jupyter-notebook 6.4.5, nbclient 0.5.3, nbconvert 0.5.3, python 3.9.7, tsai 0.2.25, fastai 2.5.3, fastcore 1.3.27, torch 1.10.2, psutil 5.8.0, pandas 1.3.4, numpy 1.19.3
- **Hardware**. The implementation and test runs were conducted on a MacBook Pro (16-inch, 2019). With a 2,6 GHz 6-Core Intel Core i7 processor. The memory specification 16 GB 2667 MHz DDR4. Browser was Google Chrome with version 100.0.4896.127 (Official Build) (x86_64).

3.4.1 Training and Validation

Team A and Team B both have each a dataset containing readiness to play and other wellness parameters from 2020 and 2021. Only the readiness data was extracted and used in this research. The training to testing ratio for the dataset is 80/20. Empirical research has shown that the range 70-80% for training and 20-30% for testing delivers the most accurate results [13]. Gholamy et al. [13] have mathematically proven that 80/20 test/training split is the most optimal ratio. The model was tested with the following datasets:

- Readiness to play for Team A/B for the year 2020
- Readiness to play for Team A/B for the year 2021
- Readiness to play for Team A/B for year 2020 and 2021 concatenated.

During testing and validation a loss function will measure the precision of the ML algorithm will be measured using a loss function. The loss function compares the model's output value against the real value. It then returns a loss as an output, which is a measure of how well the model performed [46].

Training on single player

The term testing with a player refers to the usage of only this player's data to train and test the model. The selection of the athlete was based on whether or not they were on the team for both years. Furthermore, there needed to be some variance in the values. This assessment was conducted out on the datasets for 2020, 2021, and a combination of the two years.

Training on entire team

A slightly different technique was used to train the entire team. The remaining data for the player was included after an amount of data corresponding to the input window for validation for the chosen player was eliminated. The remaining players' information was obtained into a data frame with two columns: date and readiness value. There are several benefits to using data from multiple years. The

first advantage is that the amount of data can help to improve the model. Another advantage may be to see if there is a connection between the player and the rest of the team.

3.4.2 Testing

Testing with different datasets requires no changes in the framework code due to the way the code is structured, as mentioned in section 3.1. The only additional step is to prepare the datasets into three different dataframes. One for year 2020, another for year 2021 and a third with both years concatenated. This has to be done once and then be passed to the method for prediction.

3.5 Summary

This chapter gave an overview of the new framework's role in PMSys, a comprehensive terminology list of the dataset specifications and an overall conceptualization. The built-in libraries in Python made it a natural choice. To avoid implementing the time series prediction from scratch, a deep learning library, Tsai[45] was used. The code structure and technical implementation was explained in details with examples from the code. The upcoming chapter will test the implemented framework and the results from the execution of the trained model.

Chapter 4

Analysis and Evaluation

In chapter 3, the technical implementation that makes up the system for predicting readiness was reviewed in order to better plan training sessions, choose the best players for the next competition, and possibly prevent injuries. In this chapter, we introduce the metrics we have used to evaluate our prediction framework based on the conducted experiments, and present our results centered around 4 research questions, which were specified in section 1.2.

4.1 Evaluation metrics

There are different methods and metrics that can be used when evaluating the performance of a regression problem. Model evaluation is of high importance in the field of Data Science, as it reflects the performance of a model with the respect of the outcome. A very common question is: how accurate is the model? But, accuracy is not an evaluation metric that can be used to evaluate a model developed for a regression problem. The reason for this is accuracy, which is a measurement of how often the model accurately predicts an outcome. For classification problems, it is rather a matter of how close to the actual value was it. There are several available metrics to measure the performance of a model. The most common ones are Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) [43].

Since the peaks are the most of interest MSE is a suitable metric for model evaluation, as it highlights the large errors more heavily than the small ones [29]. The larger the value of MSE, the further is the predictions from the actual values. Thus, a lower MSE is preferred. MSE is defined as follows:

$$MSE = \frac{1}{D}\sum_{i=1}^{D}(x_i - y_i)^2$$

where x_i is actual values, y_i is predicted values, and D is length of the predicted vector.

4.2 Results

The following sections will describe the outcomes of the experiments in relation to the 4 key research topics that were presented in section 1.2. The different configurations accompanied by the visual results and a table over the metrics are presented. Chapter 5 will discuss the key findings from the experiments and link it to use cases. The default configuration of the hyperparameters is Epoch 30, Batch

Size 5 and Shuffle False. The reason for that is during the initial experiments this was the configuration that had a balance between being overfit (very low MSE for training data) and being underfit (high MSE for validation and testdata).

4.2.1 Research Question 1: Training on Single Player vs. Team

Previous related work has proven that training on the entire team and predicting readiness for a single athlete resulted in promising results [49, 17, 20]. For this reason, it will be interesting to compare training on a single player/entire team and predicting a single player. The experiments were conducted on the datasets for Team A and Team B with the Epoch 30 and Batch Size 5. More detailed configurations are displayed in table 4.1.

Team	Year	Epoch	Batch size	Input window	Output window	Train on	Shuffle
A	2020	30	5	7, 14, 21	1	All	False
A	2020	30	5	7, 14, 21	1	One	False
B	2020	30	5	7, 14, 21	1	All	False
B	2020	30	5	7, 14, 21	1	One	False

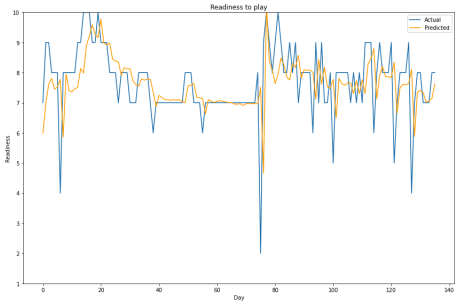
Table 4.1: An overview of the hyper-parameters and configurations for training on single player vs the entire team

The results of the experiments training on a single player and predicting for one player (see Table 4.2) showed different outcome for Team A and Team B. For Team A, the MSE value decreased from 1.64 when the input window was 7, to 1.25 when the input window was 21. Despite the fact that the MSE values for training and predicting for a single player are lower, the plot shows that it is unable to follow the curve of the real value and predicts the peaks poorly. This can be viewed in the following figures: figures 4.1b, 4.1d and 4.1f. For Team B, the MSE findings are quite the opposite. The MSE value increases as the number of input window increases. The MSE for input of 7 is 1.94, while the value for input window 21 is 2.35. Another interesting aspect is the difference in the plots compared to Team A. Figures 4.2b, 4.2d and 4.2f show that the predicted curve closely matches the actual curve compared to Team A. A reason for this could be that the values of Team B player differ from those of Team A player. Thus, the model was trained with a different range of values.

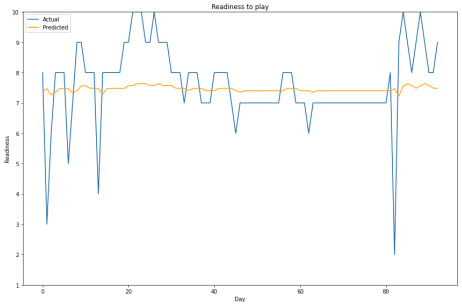
The results of the experiments training on the entire team and predicting for one player showed decreasing MSE values for Team A and the opposite for Team B. This can be caused by that the data from Team A is within a lower range than Team B. As a result, the difference between negative and positive peaks can be bigger and result in a higher MSE value. The highest MSE value for Team A was 1.54 with input window of 7 and 1.31 for an input window of 21. Team B had the highest score of 2.53 when the input window was 21 and the lowest value of 2.46 when the input window was 7. Table 4.2 provides a comprehensive summary. Both teams had plots followed the actual values more accurately than training on a single player. Peaks are also more accurately anticipated. This can be viewed in the following figures: 4.1a, 4.1c, 4.1e, 4.2a, 4.2c and 4.2e.

Team	Epoch	Batch size	Input window	Output window	Train on	Shuffle	MSE
A	30	5	7	1	Team	False	1.54
A	30	5	7	1	Player	False	1.64
A	30	5	14	1	Team	False	1.37
A	30	5	14	1	Player	False	1.49
A	30	5	21	1	Team	False	1.31
A	30	5	21	1	Player	False	1.25
B	30	5	7	1	Team	False	2.48
B	30	5	7	1	Player	False	1.97
B	30	5	14	1	Team	False	2.46
B	30	5	14	1	Player	False	2.14
B	30	5	21	1	Team	False	2.53
B	30	5	21	1	Player	False	2.35

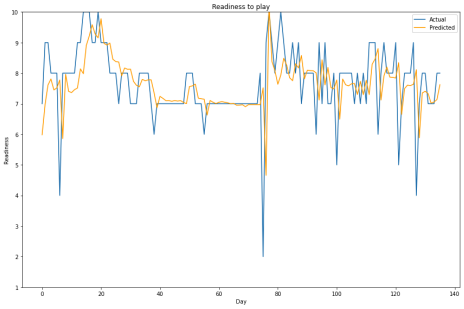
Table 4.2: Results training on single player vs training on entire team for Team A and Team B (Research Question 1).



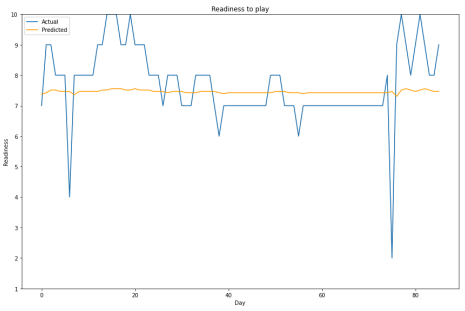
(a) Training on the entire team, Input window: 7



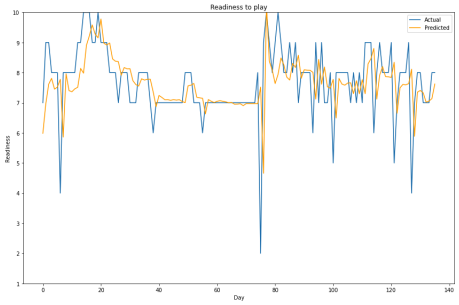
(b) Training on single player, Input window: 7



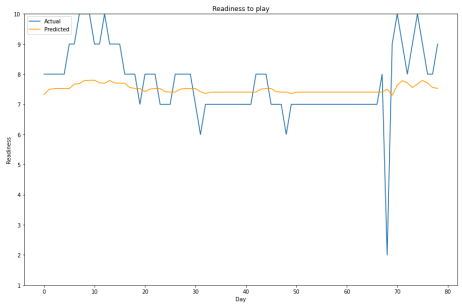
(c) Training on the entire team, Input window: 14



(d) Training on single player, Input Window 14

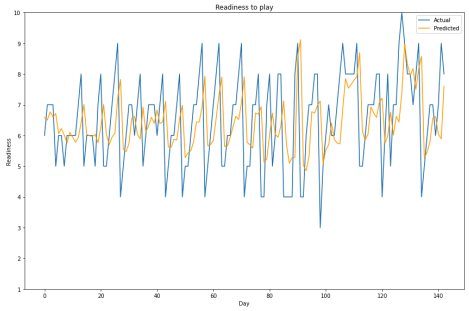


(e) Training on the entire team, Input Window 21

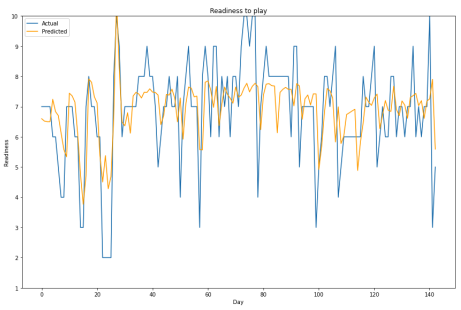


(f) Training on single player, Input Window 21

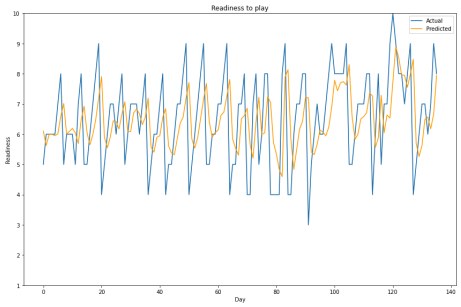
Figure 4.1: Results from training on team/single player and testing on single player from Team A, with number of epochs: 30, batch size: 5 (Research Question 1)



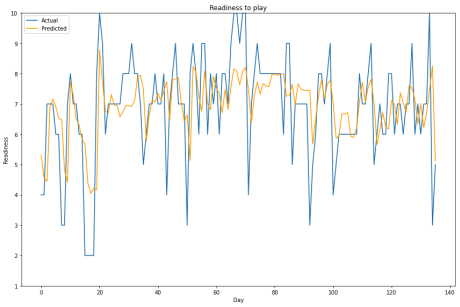
(a) Training on the entire team, Input window: 7



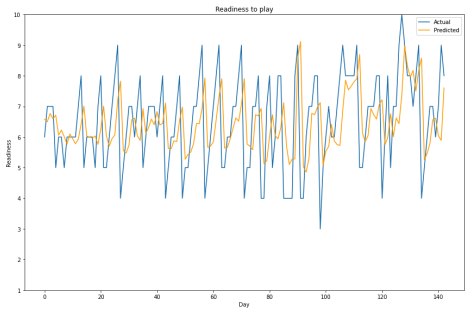
(b) Training on single player, Input window: 7



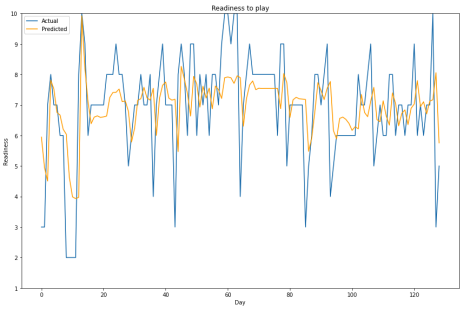
(c) Training on the entire team, Input window: 14



(d) Training on single player, Input Window 14



(e) Training on the entire team, Input Window 21



(f) Training on single player, Input Window 21

Figure 4.2: Results from training on team/single player and testing on single player from Team B, with number of epochs: 30, batch size: 5 and Ouput window: 1 (Research Question 1)

4.2.2 Research Question 2: Input and Output Window Sizes

To investigate if the input window and output window size affects the predictions, a different range of input window and output window was utilized. These experiments used a period of two years to train on the entire team and predict the readiness for one athlete. 7, 14 and 21 days were used as input windows, and 1 and 7 days were used as output window. This means that when the output window is 7, day 1...7 is predicted and with an average MSE calculated. See table 4.3 for a full overview of all parameters for the current experiment.

The results of the experiments training on the entire team and predicting one player with different input and output sizes showed the following:

Team	Epoch	Batch size	Input window	Output window	Train on	Shuffle
A	30	5	7, 14, 21	1,7	All	False
B	30	5	7, 14, 21	1,7	All	False

Table 4.3: An overview of the hyper-parameters and configurations for different input and output window sizes.

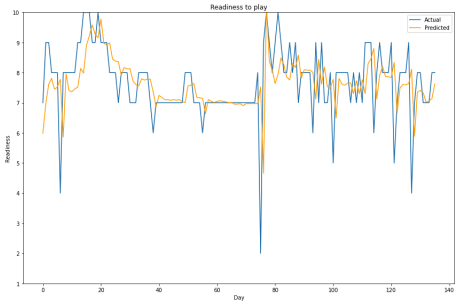
For team A, modifying the output window size to 7 resulted in an increase in the MSE values. Predicting day 1 resulted in a MSE of 1.54 which is identical for the results of predicting 1 day. The MSE value increases with the day predicted. For an input window of 7 and an output window of 7, the MSE is 1.54 for the first day and day 7 is 1.57. The same applies for the input window of 14 and 21. See table 4.4 for a full overview of all parameters for the current experiment. The increase in the deviation between the actual value and the predicted value day 1 and day 3 is visible in sub-figure 4.3b, 4.3d and 4.3f. To compare the configuration, see left column in figure 4.3.

For team B, output window size of 7 resulted in similar results as team A. With each predicted day that passes, the MSE rises. Predicting 7 days ahead when trained on 7 historical days, the MSE was lowest on day 1 with a value of 1.55 vs day 7 with a value of 2.55. That is a significant increase that is also visible in how the negative and positive peaks are deviating from the curve for actual values. When comparing the subfigures on the left and right in figure 4.4, this is clear.

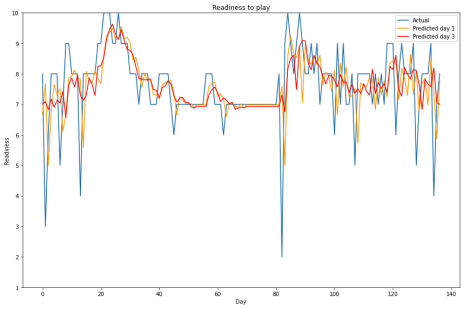
Except for the graph of 4.5a, the MSE value increases for each day predicted (see figure 4.5 for details). The reason behind the increasing MSE values for each day that passes is the increasing inaccuracy due to readiness being a continuous parameter, meaning the days before the predicted day are important. If the peaks were more periodic, the predictions for further days in the future would be able to predict the upcoming week without knowing the data of the leading week.

Team	Year	Epoch	Batch size	Input window	Output window	Train on	Shuffle	MSE
A	2020	30	5	7	1	Team	False	1.54
A	2020	30	5	14	1	Team	False	1.37
A	2020	30	5	21	1	Team	False	1.31
A	2021	30	5	7	7	Team	False	1.54-1.57
A	2021	30	5	14	7	Team	False	1.51-1.64
A	2021	30	5	21	7	Team	False	1.87-2.61
B	2020	30	5	7	1	Team	False	2.48
B	2020	30	5	14	1	Team	False	2.46
B	2020	30	5	21	1	Team	False	2.53
B	2021	30	5	7	7	Team	False	1.55-2.55
B	2021	30	5	14	7	Team	False	1.62-1.56
B	2021	30	5	21	7	Team	False	1.49-2.82

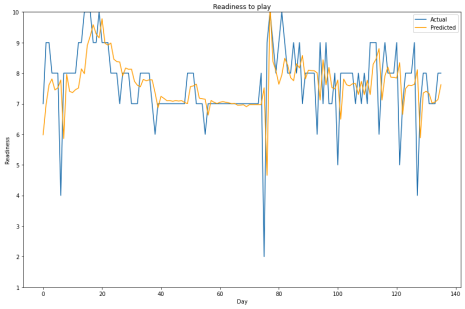
Table 4.4: Results training with different input and output window team for Team A and Team B (Research Question 2).



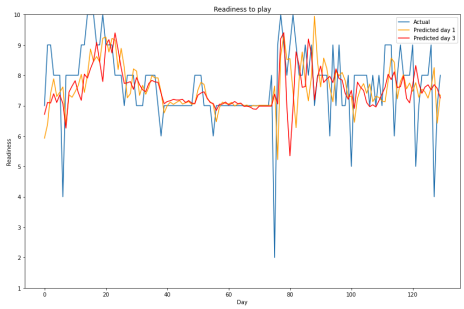
(a) Training on the entire team, Input window: 7, Output window: 1



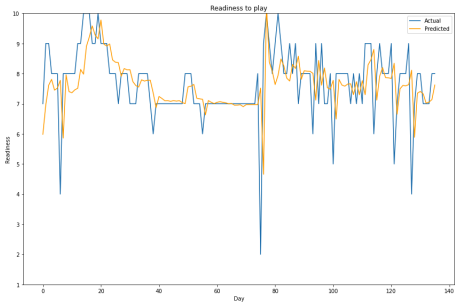
(b) Training on single player, Input window: 7, Output window: 7



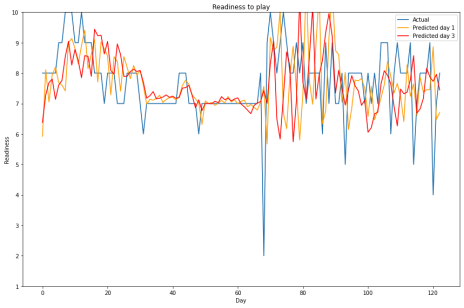
(c) Training on the entire team, Input window: 14 and Output window: 1



(d) Training on single player, Input Window 14 and Output window 7

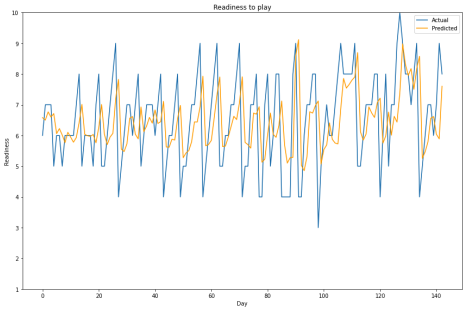


(e) Training on the entire team, Input Window 21 and Output window 1



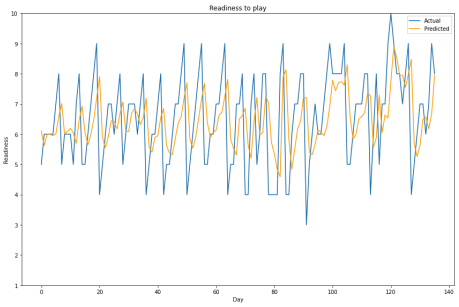
(f) Training on single player, Input Window 21 and Output window 7

Figure 4.3: Results from training with different input and output window sizes, for Team A, with number of epochs: 30 and batch size: 5 (Research Question 2)

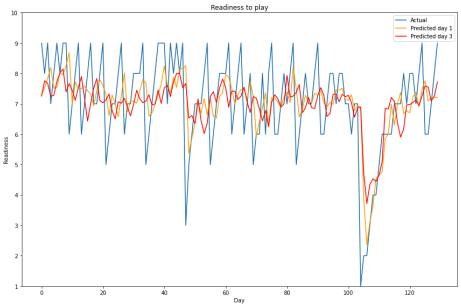


(a) Training on the entire team, Input window: 7, Output window: 1

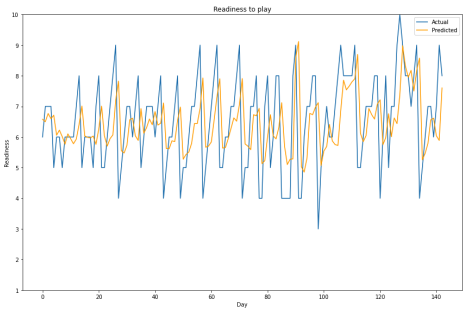
(b) Training on single player, Input window: 7, Output window: 7



(c) Training on the entire team, Input window: 14 and Output window: 1



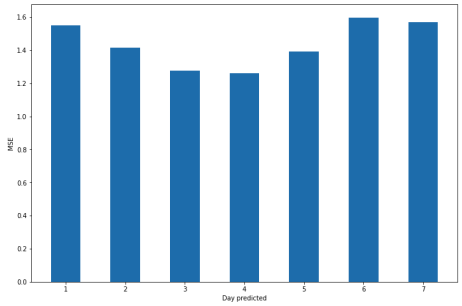
(d) Training on single player, Input Window 14 and Output window 7



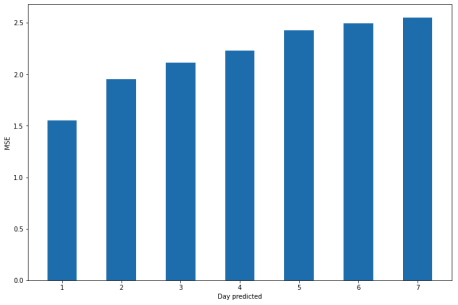
(e) Training on the entire team, Input Window 21 and Output window 1

(f) Training on single player, Input Window 21 and Output window 7

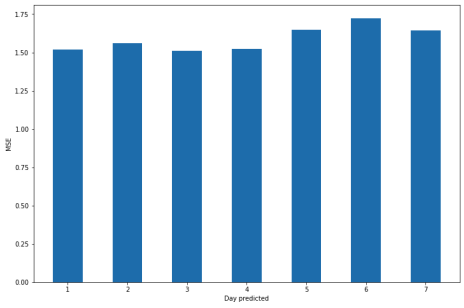
Figure 4.4: Results from training with different input and output sizes, for Team B, with number of epochs: 30 and batch size: 5 (Research Question 2)



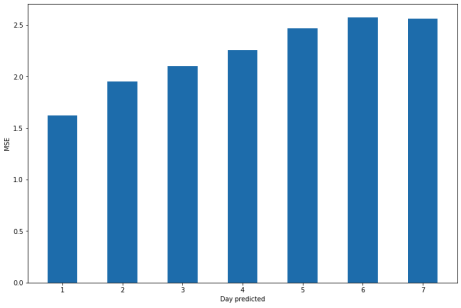
(a) Training on the entire team A, Input window: 7, Output window: 7



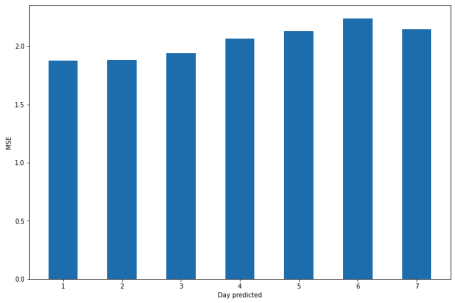
(b) Training on the entire team B, Input window: 7, Output window: 7



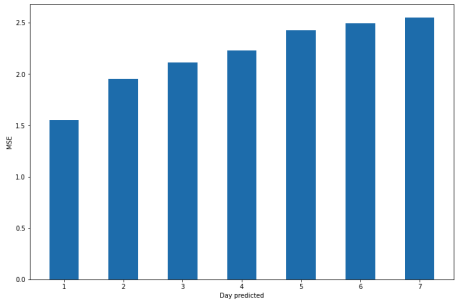
(c) Training on the entire team A, Input window: 14 and Output window: 7



(d) Training the entire team B, Input Window 14 and Output window 7



(e) Training on the entire team A, Input Window 21 and Output window 7



(f) Training on the entire team B, Input Window 21 and Output window 7

Figure 4.5: MSE results from training on the entire team and testing on single player from Team A and B, with number of epochs: 30 and batch size: 5

4.2.3 Research Question 3: Influence of Training Dataset Size

A higher number of dataset entries is known to increase the accuracy of a model and promote generalization [2]. Now, we consider how the prediction would perform if less data were used compared to a higher amount of data. The purpose of this experiment is to find out if the size of the dataset has a significant impact on the result. To do this, an athlete from each team is trained on different amounts of data. The size of the dataset will vary from 1-2 years. First, it is trained on the individual player and predicts readiness for this. Then, it is trained on the whole team and tested on the player. For additional configurations, see table 4.5.

The results of the experiments training on a single player and predicting one player with one year of data resulted in a higher MSE than training on the entire

Team	Year	Epoch	Batch size	Input window	Output window	Train on	Shuffle
A	2020	30	5	7	1	All	False
A	Two years	30	5	7	1	One	False
A	2020	30	5	7	1	All	False
A	Two years	30	5	7	1	One	False
B	2021	30	5	7	1	All	False
B	Two years	30	5	7	1	One	False
B	2021	30	5	7	1	All	False
B	Two years	30	5	7	1	One	False

Table 4.5: An overview of the hyper-parameters and configurations trained on different training dataset size.

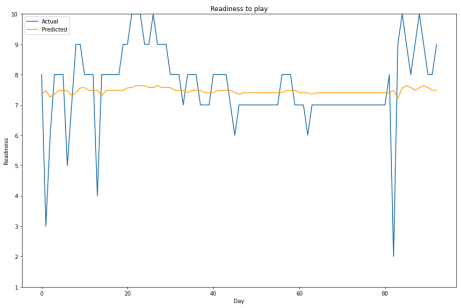
team and predicting for a single player with a year of data. For Team A, the MSE was 1.54 when training on all and 1.64 when training on one. However, the plots reveal that the actual curve is more closely followed by the predicted curve when the datasize is two years instead of one. This is especially clear to see when comparing the left and right subfigures in figure 4.6. The model is able to predict the readiness with higher precision with an increased amount of data.

The results of the experiments training on the entire team and predicting one player with one year of data resulted in a lower MSE than training on a single player. Training on all with one year of data for Team B resulted in an MSE of 1.60. Compared to training on one, the MSE was 1.77. Figure 4.6d and 4.7d convey that the model is able to follow the peaks more closely. This is especially true for the negative peaks. Thus, training on all and predicting a single player results in higher precision in following the peaks and a lower MSE.

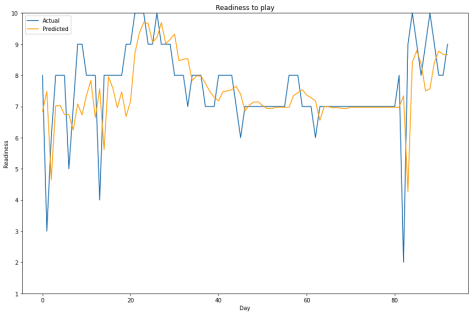
Both the graphs and the MSE values convey that the model performs best with an increased amount of data. Training on the entire team with one year of data performed better than training on a single player with 2 years of data. Training on the entire team with two years of data best followed the peaks, but had a higher MSE than the same case with one year of data.

Team	Year	Epoch	Batch size	Input window	Output window	Train on	Shuffle	MSE
A	2020	30	5	7	1	Team	False	1.54
A	Both	30	5	7	1	Team	False	1.63
A	2020	30	5	7	1	Player	False	1.64
A	Both	30	5	7	1	Player	False	1.63
B	2021	30	5	7	1	Team	False	1.60
B	Both	30	5	7	1	Team	False	1.91
B	2021	30	5	7	1	Player	False	1.77
B	Both	30	5	7	1	Player	False	2.14

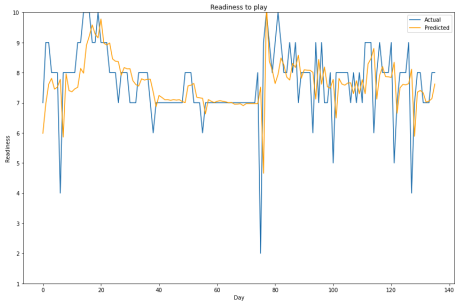
Table 4.6: Results training with different input and output window team for Team B (Research Question 3).



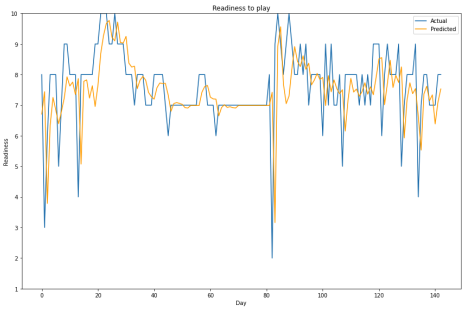
(a) Training on single player with 1 year of data, Input window: 7, Output window: 1



(b) Training on single player with 2 years of data, Input window: 7, Output window: 1

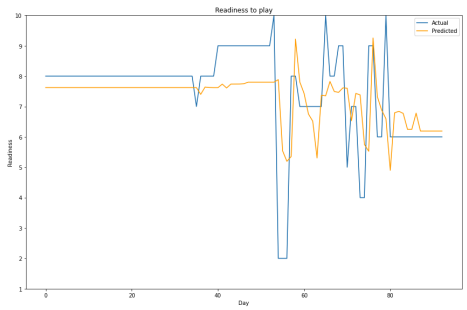


(c) Training on the entire team with 1 year of data, Input window: 14 and Output window: 1

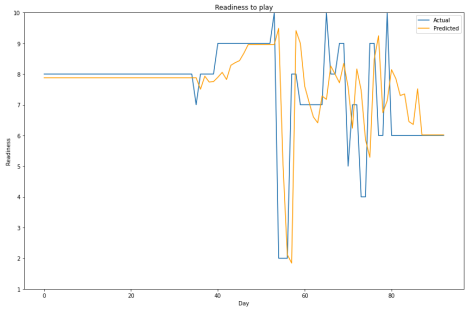


(d) Training on entire team with 2 years of data, Input Window 14 and Output window 1

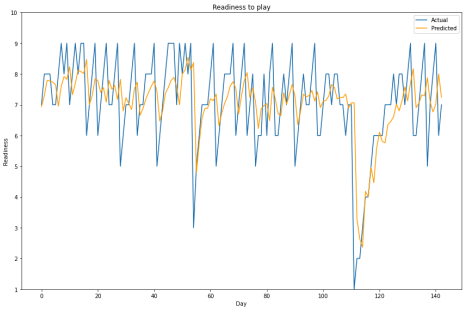
Figure 4.6: Results from training on different dataset sizes, for Team A, with number of epochs: 30 and batch size: 5 (Research Question 3)



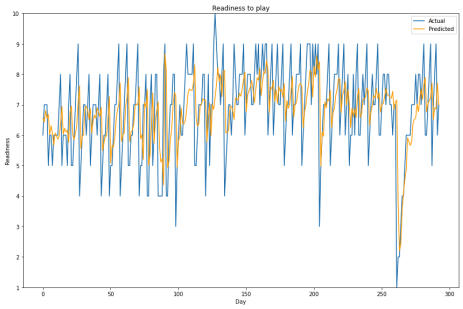
(a) Training on single player with 1 year of data, Input window: 7, Output window: 1



(b) Training on single player with 2 years of data, Input window: 7, Output window: 1



(c) Training on the entire team with 1 year of data, Input window: 7 and Output window: 1



(d) Training on entire team with 2 years of data, Input Window 14 and Output window 1

Figure 4.7: Results from training on different dataset sizes, for Team B, with number of epochs: 30 and batch size: 5 (Research Question 3)

4.2.4 Research Question 4: Influence of Hyperparameters

Model optimization depends on what is called hyper-parameters, which, in this case include epoch, batch size and shuffle as described in section 3.1. This section will show the impact of the different hyper-parameters when modified. For this experiment, a single player from a single team is picked. The hyper-parameters are modified and the model is trained on the entire team and predicted on a player. For additional configurations, see table 4.7.

Team	Epoch	Batch size	Input window	Output window	Train on	Shuffle
A	30, 40, 50	5, 25	7	1	All	False, True
A	30, 40, 50	5, 25	7	1	One	False, True

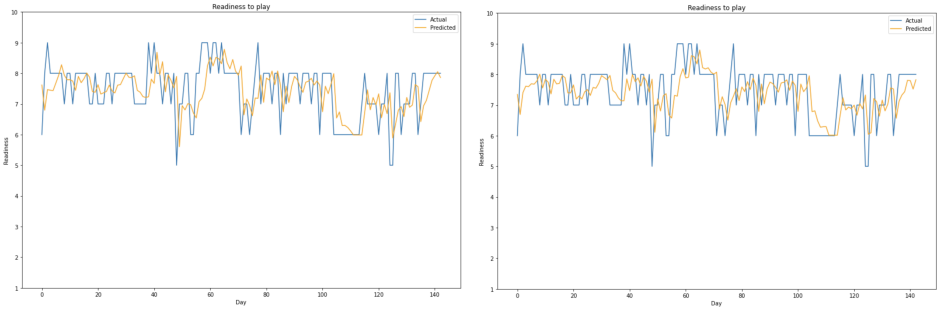
Table 4.7: An overview of the hyper-parameters and configurations conducted on training with modified hyper-parameters

The experiments showed that enabling shuffle led in a reduced MSE while training the entire team and predicting for one player with different hyper-parameters. The plots of actual and predicted values, however, were almost identical. For all six cases, the peaks were quite near to the actual values. The

only significant component that reduced the MSE value was changing shuffle from false to true, however it was not of high importance.

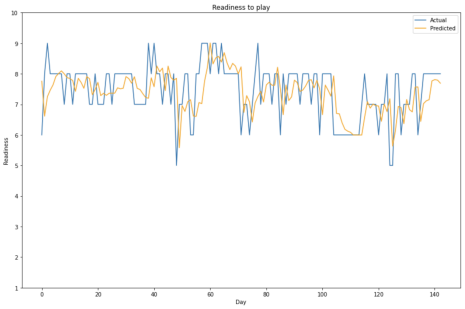
Team	Epoch	Batch size	Input window	Output window	Train on	Shuffle	MSE
A	30	5	7	1	Team	True	0.68
A	30	5	7	1	Team	False	0.72
A	40	5	7	1	Team	True	0.72
A	40	5	7	1	Team	False	0.73
A	30	25	7	1	Team	True	0.68
A	30	25	7	1	Team	False	0.69
A	50	5	7	1	Team	True	0.86
A	50	5	7	1	Team	False	0.76
A	50	25	7	1	Team	True	0.70
A	50	25	7	1	Team	False	0.73

Table 4.8: Results from training on entire team with different hyper-parameters (Research Question 4).

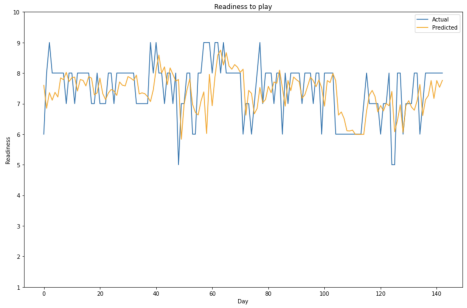


(a) Epoch 30, Batch size 5, Shuffle false.

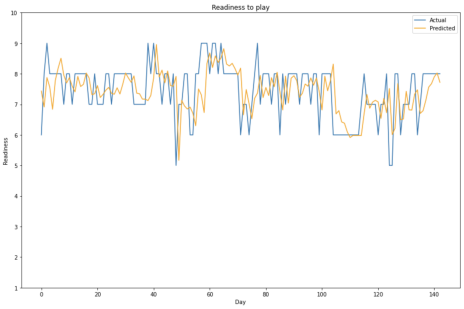
(b) Epoch 30, Batch size 5, Shuffle true.



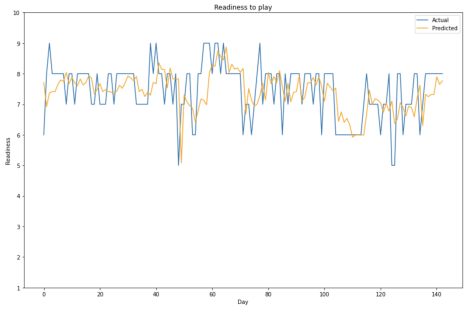
(c) Epoch 40, Batch size 5, Shuffle false.



(d) Epoch 40, Batch size 5, Shuffle true.



(e) Epoch 50, Batch size 25, Shuffle false.



(f) Epoch 50, Batch size 25, Shuffle true.

Figure 4.8: Results from training with different hyperparameters, for Team A, with Input window 7 and output window 1 (Research Question 4).

4.3 Summary

In this chapter, we presented our results regarding the 4 research questions we addressed, namely (1) we looked at how training on a single player vs training on the entire team affected the results, (2) we looked at the impact of different input and output sizes, (3) we looked at the influence of the dataset size and compared one year of tranining data with two years of training data for both single player and entire team, (4) Finally we looked at what influence the different hyperparameters such as Epoch, Batch Size, and Shuffle had on the predicted values. We found out that (1) Training on the entire team and predicting for a single player resulted in more accurate overall results for Team A, where player consistency was higher, whereas overall accuracy was lower for Team B, where player consistency was lower. However, the peaks were more correctly predicted compared to when training on a single player for both teams. (2) The inaccuracy increases with each predicted day in the future. This means that an output window of 1 followed the

peaks better than an output window of 7. Increasing the input window did not seem to help with the accuracy of the predicted values, (3) We observed that more data does not necessarily help prediction accuracy, neither for models trained on the complete team nor on individual players. However, training on the entire team with one year of data outperformed training on a single player with two years of data. This means that a shorter period of collected data for a team can be more beneficial than data over a longer period. (4) The hyper-parameters were changed, but no significant deviations occurred. The shuffling appeared to be the only property of importance, where turning it on proved slightly more beneficial.

Chapter 5

Discussion

In this work, our main research question to investigate was: *Can we predict readiness for elite female soccer athletes using machine learning on data collected using an athlete monitoring system?* Moreover, in the previous chapter, we proved that there is a potential in order to apply machine learning to this kind of time-series predictions. In this section, we discuss our results and look back on what has been achieved throughout the project.

5.1 Insights

In chapter 4, we discovered that predicting a single player while training the entire team caused the predicted values to closely mimic the negative and positive peaks. This means that our approach can still be used in a team-agnostic way, as long as the motivation is more focused on peak prediction than overall prediction. Training on the entire team can benefit greatly in predicting the values for a single player, which is of value when we want to adjust training schedule around the negative and positive peaks.

In our experiments, we looked at several aspects related to dataset, training, and hyper-parameters. Our results convey that training on the team and predicting single player showed better results in term of peak following. The inaccuracy increases with each predicted day in the future. More data does not necessarily help with the accuracy of the predicted values. No significant changes occurred when the hyper-parameters were modified. This chapter is going to further elaborate the findings with accompanying explanation. Limitations on the results will also be discussed, and examples of uses for this research will be suggested.

When the output window is larger than one, the prediction error grows with each passing day. Increasing the input window had no effect. We believe this is related to the lack of periodicity in the data, and it demands further investigation using different datasets, as well as multivariate prediction utilizing the other wellness parameters included in table 3.2.

Furthermore, predicting a single player with a single year of data outperformed training on a single player with two years of data. This indicates that training on a more recent dataset is better than a longer, more outdated amount of data. For team, this is advantageous in a variety of situations, including when there is minimal data at the start of a training season or when a new player joins the club.

Finally, changing the hyper-parameters had no discernible effect on the

predictions. However, turning on Shuffle improved the results a bit, and if that margin is something the teams may take advantage of, then this should be considered as the default value. This leaves room for the teams to experiment with different hyper-parameters on the go to see what suits them the most. As a starting point the hyper-parameter setup with the least amount of resource consumption is preferred.

5.2 Potential Use Cases

Our results show that an athlete training system could be used in the following ways:

- If the main area of interest is the peaks, then use a team-based model. Played-based models are better suited if the overall accuracy is the main concern.
- Daily predictions deliver the best performance. Our approach can be used for daily predictions, which means that a team could possibly run models with a batch job for a fixed period of each month/week/day for the upcoming day. Weekly predictions are also possible to if you have more recent data.
- Recent collected data gives more accurate results compared to a higher but older quantity of data. Datasets with entries from the entire team on a shorter period is better than a longer period of data for a single player. This implies that after a short period of data has been collected for a team, accurate predictions may be made.
- The hyper-parameters, such as Epoch, Batch Size and Shuffle, can be optimized in a customized fashion to adjust to the needs of the team given their training data etc. We already propose a starting point of the following Epoch: 30, Batch Size: 5 and Shuffle: True. This is a configuration that uses very little resources. Higher values use more resources and have yet to demonstrate that they are worthwhile.

5.3 Limitations

This section will discuss the limitations that emerged during the thesis.

- **Dataset size**: The findings in section 4.2.3 revealed that the size of the dataset has a high impact on the results when training on the entire team. A higher number of athletes that the model trains on, results in predicted values close to the actual values and a lower MSE. Since Team B had more players both years, the results for Team A might have been affected. The year 2021, Team A had 23 players and Team B had 28 players. The year 2020 Team A had 15 players and Team B had 24 players. There is a significant amount of data error, which may have affected the results.
- **Missing data**: The datasets for Team A and Team B suffer from a lack of reported readiness values. This can be easily viewed in table 3.1. Since this is an aspect of high importance in the field of data analytics, it is further elaborated in section 5.4.
- **Weak periodicity**. The parameter of choice, readiness is weakly periodic and highly different way of reporting across teams. This increases the difficulty to predict readiness for a longer output window with a high accuracy.

- **Univariate prediction**: Readiness is a complicated parameter that is influenced [17] by other wellness factors including sleep duration, stress, and mood. This study reveals that predicting readiness using only one variable is difficult, and that a multivariate solution may be more accurate.
- **Lack of documentation**: For RNN-related tasks, Tsai [45] looks to be a novel, intriguing, and powerful framework. Unfortunately, it is also far too premature, and the documentation is woefully inadequate. As a result, the implementations, as well as the motivations behind them, are not well understood. This is notably lacking in the LSTMPlus documentation but is recurring in the rest of the documentation. In future iterations, different libraries with better documentation could be considered by the teams, as Tsai as of 2022 has the earlier mentioned shortcomings. Tsai is, nevertheless, adequate at the moment.
- **Time**: Running experiments with different permutations of hyper-parameters would have been desirable. Unfortunately, conducting experiments with different configurations is really time consuming and not possible in the time-frame of this thesis.

5.4 Missing Data Problem

The definition of missing data is the lack of recorded data where a value should exist. This is a very common problem in the field of data science [22]. Missing data is a prevalent problem in nearly all research, and it can have a vast impact on the conclusions that can be derived [15]. There are several tactics to handle missing data problem, such as removing the entries without values, backward fill, forward fill and a combination of the latter. This thesis uses the backward fill which means that the closest historical value is used. The reason for that is there were periods in the datasets that had historical values, but lacked future entries. Thus, it made sense to make use of the historical values we had. Missing data is a research question on its own and thus out of scope for this thesis. The problems that occur due to missing data are opportunities for researches to develop adequate techniques to overcome them [31].

5.5 Privacy and Athlete Anonymity

With regards of privacy and athlete anonymity there are two main areas of concern:

- **In the cloud**. The PMSys framework uses a third party service, Amazon Web Services (AWS) [3], for storing their data. AWS provides full control of where the data is stored, access management and what resources your organization is using at any given time. In addition, the security technology AWS rely on is by familiar solution providers known to be fully trusted. All the information that flows through the AWS global network is automatically encrypted at the physical layer before it leaves their facilities [3]. The backend in PMSys maps each athlete to an unique id during persistence of the data. As mentioned in section 2.3.2, the DSU is kept separated from other logic. This means that the endpoints that the frontend components use (mobile application and trainer portal) only work with the user id and thus preserve the user anonymity and privacy.

- **In the team**. The privacy of the athletes are highly valued and thus each following privacy-preserving guidelines [17] as mentioned in section 3.2. Each trainer has the access to view the reported metric for each respective player on their team with and without identified names. This introduces the challenge of bias in the subjective self-reported data. Since the data is not anonymous the players may feel the urge to influence the reported wellness metrics in order to not be put on a different regime than the other teammates. Players who report lower scores on readiness to play, less sleep or bad mood might fear that they will be benched, and thus report higher scores than they actually are. Predicting the team's future performance for the following week or day will be of higher difficulty if the data supplied is incorrect. To avoid presenting a false picture, there are some steps needed to prevent athletes from experiencing psychological stress as a result of feeling as less of a performer or excluded during training. In addition, there must be enough psychological security in the team for players to feel that they can report honest data for their own good. An athlete motivated by fear or failure may report inaccurate data.

  If this approach is actively used, you will see significant differences in how players feel. If a player is over-trained then the coach should bench the proper player, thus one must know who this pertains to. As a result, players will be under a lot of strain and will get fixated on the stated subjective information. Players may no longer provide an honest image of how they feel for fear of being benched or having a lower training frequency than the rest of the squad.

  An option would be to make the athlete identification completely anonymous. This introduces a different issue: how do you approach the ones affected? If the predictions show that one player needs rest, but the rest of the team feels good, then you do not know which player it is. Thus, it can either be the case that nothing is done about it since you do not know who it is. Or, you can scale down the intensity of training for the whole team. This means that the team does not train at the optimal intensity, which in turn can affect their performance negatively. In a match context, this can be the difference between win and loss.

5.6 Summary

In this chapter, we discussed various aspects of our results and experiences, including comparison of other models and potential use cases. The main takeaways for teams considering using a data analysis framework is to evaluate what their main area of interests is. Is it predicting the negative and positive peaks or predicting the overall accuracy? The length of the input window has little impact on the relevance of the outcome, and even a week of input window is adequate. Peak predictions are improved by a team-based model, although overall accuracy is better served by single-player models. Daily predictions are more accurate than multi-day predictions unless the data is recent. Setting Shuffle to True can enhance the MSE.

Chapter 6

Conclusion

6.1 Summary and Main Contributions

Soccer is by far the most popular team sport, and it continues to grow in popularity. Despite soccer being a male dominant sport, the popularity and professionalism of female soccer has increased remarkably. Female athletes are now employed on both professional and semi-professional level [6]. Injury prevention is a key concern in the longevity of an athlete's carrier. It can be quiet costly for an athlete club, if not properly monitored. With the help of ML, we can automate the analysis and prediction process, allow for personalized training and provide valuable insights that the bare eye cannot reveal. Due to the increased training and competition demands, ML techniques are being enforced to in decision making to optimize performance. In other domains of study, such as psychology, self reporting is a frequently utilized and acknowledged tool for achieving relevant insights [49].

In this thesis, we developed framework and evaluated the self reported wellness metric, *readiness*, with regards to soccer athlete performance metrics using ML. This was conducted using a dataset containing metric collected over 2 years from 2 clubs in the Norwegian elite women's soccer league. The programming language Python and the Deep Learning Library, Tsai, are used in the technical implementation. The end result is a framework that can be simply integrated into an existing athlete monitoring system as a future step. The experiments were carried out with a set of permutations of hyperparameters and different size of the training and test data.

6.1.1 Answering the Research Questions

We established certain aims and an initial research question at the start of this thesis, and we will illustrate how we approached and solved the task. The overall research question was introduced in section 1.1: *Can we predict readiness for elite female soccer athletes using machine learning on data collected using an athlete monitoring system?*

We broke down the issue statement into four sub-questions, which we addressed and answered.

RQ1. Is it more accurate to predict a player's readiness using data from an individual player or team based data? Team predictions works better peak predictions, and we showed that this conclusion is consistent across multiple teams. A 7 day input window and an output window of 1 with an epoch of 30 and batch size of 5, resulted in a MSE value of 1.54 for team-

based approach vs. 1.64. This is particuraly noticable when comparing left subfigures with right subfigures in figure 4.1.

RQ3. Does the training dataset size have an impact on the results, and is a year or two the best for accurate predictions? One day predictions are more accurate than a several day prediction. The reason for that is the increasing inaccuracy for each passing day. 1 day prediction returned an MSE value 1.31 compared to a 7 day prediction resulted in a MSE value 1.51 (see table 4.4 for details). The increasing deviation from the actual values is visable in figure 4.4.

RQ3. Does the training dataset size have an impact on the results, and is a year or two the best for accurate predictions? A shorter period of data collected from a team is more valuable and procduces more accurate predictions than a higher amount of data from a single player. Less, but more recent data performs better than a longer period of more data. Traning with a year of data on Team A resulted in an MSE of 1.60, compared to two years of data and trained on a single player the MSE was 1.77.

RQ4. What permutation of the hyperparameters result in more accurate results? Modification of hyperparameters did not result in any significant results. Changing Shuffle to true resulted in the MSE value lowering from 0.72 to 0.68. In the study by Kulakou [20], he showed that the multivariate experiments responded well to the adjustment of hyper parameters. Further experimentation should be conducted with several permutations of the hyperparameters.

The points above show how we addressed and answered all the different sub-questions derived from the overall research question. Thus, the system we have developed and the experimental results show what machine learning, i.e., LSTM in our case, can be used to predict future time-series values, and in particular the future *readiness to play* of a soccer player.

A system like this can have a huge impact in the field of sports science and computer science. There will opportunities for further development of state-of-the-art platforms and athlete monitoring gears tailored for each sports to gather accurate and valuable information. By allowing AI and ML to process and convey information that cannot be easily detected, evidence based decisions can be made to prevent overuse and injuries.

6.1.2 Other Contributions

Other scientific contributions, apart from but directly related to this thesis, are as follows:

- **Source code:** All the software that has been used for this work is publicly accessible under the repository: https://github.com/simula/pmsys
- **Colab notebooks:** In order to promote reproducibility, we provide our software in the form of directly executable Google Colab notebooks under our public repository. On different platforms, setting up an interactive Python environment might be a very time consuming task. Google Colab enables users to execute interactive Python notebooks in the browser, removing platform dependencies.
- **Paper 1:** Presentation abstract titled "Soccer Athlete Performance Prediction using Time Series Analysis" (accepted to appear at the NORA Annual Conference 2022), details available in section A.1.

- **Paper 2:** Dataset paper titled "SoccerMon: A Large-Scale Multivariate Dataset of Soccer Athlete Health, Performance, Training Load, and Position Monitoring" (to be submitted to Nature Scientific Data in May 2022), details available in section A.2.
- **Paper 3:** Journal abstract titled "Exploration of Different Time Series Models for Soccer Athlete Performance Prediction" (submitted to ITISE 2022), details available in section A.3.

6.2 Future Work

There are several experiments that would have been carried out, if time allowed. Here are some of the suggestions that we believe will add the most value to this framework.

- Readiness is a complex parameter, based on the overlay of multiple other metrics. This is also why PmSys collects a multitude of other wellness metrics, namely stress, mood, sleep duration to name a few. See table 3.2 for more details. For future iterations, a multivariate time series analysis can yield more detailed understanding of how readiness is influenced, and how it changes over time. This can actually yield periodicity, and allow for the use of simpler time series tools. In the findings by [20], the changes of hyperparameters had the most effect on multivariate predictions. Investigate if this is the case.
- Investigate the contextual relevance of the samples. What effects does it have on the predictions when using off-season vs on-season, weekday vs weekend, match day vs regular training?
- Investigate other Deep Learning libraries and pick a mature and well documented one. Tsai is under continuous development and it is possible that Tsai is still sufficient and better documented in the future.
- The user interface for menstrual cycle collection in PMSys as can be observed in figure 2.2. In a study conducted by Julian et al. [38], they concluded that an athletes endurance during the Luteal Phase was noticeably reduced. Thus, menstrual cycle is a wellness parameter that can have tremendous impact on readiness to play, depending on which phase in the cycle the female athlete is currently in
- Consider solving the missing data problem with different approached. Try with forward fill, interpolating or removing NAN values. Another alternative is to populate the field of the days with an average value of the previous days or week.
- This framework have dedicated a stage in the pipeline to offer developers continuous feedback on the performance of the designed system. As part of the automated testing deployment, this allows for regular model updates.

Bibliography

[1] Peter J. Denning (Chairman), Douglas E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner and Paul R. Young. 'Computing as a Discipline'. In: (1989). DOI: https://doi.org/10.1145/63238.632395. URL: https://dl.acm.org/doi/10.1145/63238.63239.

[2] Ajiboye Abdulraheeml and Ruzaini Abdullah Arshah. 'Evaluating the Effect of Dataset Size on Predictive Model Using Supervised Learning Technique'. In: (2015). URL: https://www.researchgate.net/publication/284371947_Evaluating_the_Effect_of_Dataset_Size_on_Predictive_Model_Using_Supervised_Learning_Technique.

[3] *Amazon Web Services*. 2022. URL: https://aws.amazon.com/security/ (visited on 11/05/2022).

[4] Orkun Baloglu, Samir Q Latifi and Aziz Nazha. In: *What is machine learning?* 2021. DOI: 10.1136/archdischild-2020-319415. URL: https://ep-bmj-com.ezproxy.oslomet.no/content/early/2021/02/08/archdischild-2020-319415.

[5] Jason Brownlee. *Deep Learning for Time Series Forecasting*. 2019.

[6] Naomi Datson, Andrew Hulton, Helena Andersson, Tracy Lewis, Matthew Weston, Barry Drust and Warren Gregson. In: *Applied Physiology of Female Soccer: An Update*. 2014. DOI: 10.1007/s40279-014-0199-1. URL: https://link.springer.com/article/10.1007/s40279-014-0199-1.

[7] Eyal Eliakim, Elia Morgulev, Ronnie Lidor and Yoav Meckel. 'Estimation of injury costs: financial damage of English Premier League teams' underachievement due to injuries'. In: (2020). DOI: 10.1136/bmjsem-2019-000675. URL: https://bmjopensem.bmj.com/content/6/1/e000675.

[8] *fastai*. 2022. URL: https://github.com/fastai/fastai (visited on 10/04/2022).

[9] FFRC. *Female Football Research Centre*. https://uit.no/research/ffrc. (Accessed on 21.01.2021).

[10] *Fifa*. 2022. URL: https://www.fifa.com/tournaments/mens/worldcup/2018russia/media-releases/more-than-half-the-world-watched-record-breaking-2018-world-cup (visited on 11/05/2022).

[11] *Forzsys*. 2022. URL: https://forzasys.com/pmSys.html (visited on 12/05/2022).

[12] C W Fuller, J Ekstrand, A Junge, T E Andersen, R Bahr, J Dvorak, M Hägglund, P McCrory and W H Meeuwisse. 'Consensus statement on injury definitions and data collection procedures in studies of football (soccer) injuries'. In: (2011). DOI: 10.1136/bjsm.2005.025270. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2491990/.

[13] Afshin Gholamy, Vladik Kreinovich and Olga Kosheleva. *Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation*. Tech. rep. 2018. URL: https://scholarworks.utep.edu/cgi/viewcontent.cgi?article=2202&context=cs_techrep (visited on 11/04/2022).

[14] Ian Goodfellow, Yoshua Bengio and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[15] John W. Graham. In: *Missing Data Analysis:Making It Work in the Real World*. 2009. DOI: 10.1146/annurev.psych.58.110405.085530. URL: https://doi.org/10.1146/annurev.psych.58.110405.085530.

[16] Sepp Hochreiter and Jürgen Schmidhuber. 'Long Short-Term Memory'. In: (1997). DOI: 10.1162/neco.1997.9.8.1735. URL: https://ieeexplore.ieee.org/abstract/document/6795963.

[17] Håvard D. Johansen, Dag Johansen, Tomas Kupka, Michael A. Riegler and Pål Halvorsen. 'Scalable Infrastructure for Efficient Real-Time Sports Analytics'. In: *Companion Publication of the 2020 International Conference on Multimodal Interaction*. ICMI '20 Companion. Virtual Event, Netherlands: Association for Computing Machinery, 2020, pp. 230–234. ISBN: 9781450380027. DOI: 10.1145/3395035.3425300. URL: https://doi.org/10.1145/3395035.3425300.

[18] Bandyopadhyay K. *Legacies of Great Men in World Soccer: Heroes, Icons*. 2017.

[19] Donald T. Kirkendall. 'Evolution of soccer as a research topic'. In: (2020). DOI: https://doi.org/10.1016/j.pcad.2020.06.011. URL: https://www.sciencedirect.com/science/article/abs/pii/S0033062020301341?via%5C%3Dihub.

[20] Siarhei Kulakou. 'Exploration of time-series models on time series data'. MA thesis. University of Oslo, Dec. 2021.

[21] Springer Nature Limited. *Scientific Data*. 2022. URL: https://www.nature.com/sdata/ (visited on 16/05/2022).

[22] BrownleeRoderick J. A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data*. 2019.

[23] *LSTM architecture Latex*. 2022. URL: https://tex.stackexchange.com/questions/432312/how-do-i-draw-an-lstm-cell-in-tikz (visited on 14/05/2022).

[24] *LSTM Framework For Univariate Time-Series Prediction*. 2021. URL: https://towardsdatascience.com/lstm-framework-for-univariate-time-series-prediction-d9e7252699e (visited on 12/05/2022).

[25] *LSTMPlus*. 2022. URL: https://timeseriesai.github.io/tsai/models.RNNPlus.html (visited on 05/04/2022).

[26] Q. Ma. In: *Comparison of ARIMA, ANN and LSTM for Stock Price Prediction*. 2021. DOI: 10.1051/e3sconf/202021801026.

[27] Esben Elholm Madsen, Tina Hansen, Sidsel Damsgaard Thomsen, Jeppe Panduro, Georgios Ermidis, Peter Krustrup, Morten B. Randers, Carsten Hvid Larsen, Anne-Marie Elbe and Johan Wikman. 'Can psychological characteristics, football experience, and player status predict state anxiety before important matches in Danish elite-level female football players?' In: (2020). DOI: https://doi.org/10.1111/sms.13881. URL: https://onlinelibrary.wiley.com/doi/epdf/10.1111/sms.13881.

[28] Pankaj Malhotral, Lovekesh Vig, Gautam Shroff and Puneet Agarwal. 'Long Short Term Memory Networks forAnomaly Detection in Time Series'. In: (2015). URL: https://www.researchgate.net/publication/304782562_Long_Short_Term_Memory_Networks_for_Anomaly_Detection_in_Time_Series.

[29] *MSE*. 2022. URL: https://en.wikipedia.org/wiki/Mean_squared_error (visited on 05/04/2022).

[30] *NORA Annual Conference 2022*. 2022. URL: https://www.nora.ai/nora-annual-conference/annual-conference-2022/ (visited on 10/04/2022).

[31] Heru Nugroho and Kridanto Surendro. 'Missing Data Problem in Predictive Analytics'. In: (2019). URL: https://dl.acm.org/doi/10.1145/3316615.3316730.

[32] *NumPy*. 2022. URL: https://numpy.org/ (visited on 29/04/2022).

[33] Svein A. Pettersen, Håvard D. Johansen, Ivan A. M. Baptista, Pål Halvorsen and Dag Johansen. 'Quantified Soccer Using Positional Data: A Case Study'. In: *Frontiers in Physiology* 9 (2018). ISSN: 1664-042X. DOI: 10.3389/fphys.2018.00866. URL: https://www.frontiersin.org/article/10.3389/fphys.2018.00866.

[34] *PMSYS*. 2020. URL: https://forzasys.com/pmSys.html (visited on 06/04/2022).

[35] Raffaele Pugliese, Stefano Regondi and Riccardo Marini. In: *Machine learning-based approach: global trends, research directions, and regulatory standpoints*. 2021. DOI: 10.1016/j.dsm.2021.12.002. URL: https://reader.elsevier.com/reader/sd/pii/S2666764921000485.

[36] *Python*. 2022. URL: https://www.python.org/ (visited on 15/05/2022).

[37] *Pytorch*. 2022. URL: https://pytorch.org/ (visited on 10/04/2022).

[38] Julian R, Hecksteden A, Fullagar HHK and Meyer T. In: *The effects of menstrual cycle phase on physical performance in female soccer players*. 2017. DOI: 10.1371/journal.pone.0173951. URL: https://doi.org/10.1371/journal.pone.0173951.

[39] *RNN architecture Latex*. 2022. URL: https://tex.stackexchange.com/questions/494139/how-do-i-draw-a-simple-recurrent-neural-network-with-goodfellows-style (visited on 14/05/2022).

[40] Sima Siami-Namini, Neda Tavakoli and Akbar Siami Namin. In: *Missing Data Analysis:Making It Work in the Real WorldA Comparative Analysis of Forecasting Financial Time Series Using ARIMA, LSTM, and BiLSTM*. 2019. URL: https://arxiv.org/abs/1911.09512.

[41] *Supervised vs. Unsupervised Learning: What's the Difference?* 2022. URL: https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning (visited on 12/05/2022).

[42] *Supervised vsNeural Networks*. 2022. URL: https://www.ibm.com/cloud/learn/neural-networks (visited on 12/05/2022).

[43] *Time Series Forecast Error Metrics You Should Know*. 2022. URL: https://towardsdatascience.com/time-series-forecast-error-metrics-you-should-know-cc88b8c67f27s (visited on 12/05/2022).

[44] *timeseriesAI*. 2022. URL: https://github.com/timeseriesAI (visited on 11/04/2022).

[45] *Tsai*. 2022. URL: https://github.com/timeseriesAI/tsai (visited on 10/04/2022).

[46] *Understanding the 3 most common loss functions for Machine Learning Regression*. 2022. URL: https://towardsdatascience.com/understanding-the-3-most-common-loss-functions-for-machine-learning-regression-23e0ef3e14d3 (visited on 16/05/2022).

[47] *What is Deep Learning?* 2020. URL: https://machinelearningmastery.com/what-is-deep-learning/ (visited on 14/05/2022).

[48] *What Is Machine Learning?* 2022. URL: https://link.springer.com/chapter/10.1007/978-3-319-18305-3_1 (visited on 12/05/2022).

[49] Theodor Wiik, Håvard D. Johansen, Svein-Arne Pettersen, Ivan Baptista, Tomas Kupka, Dag Johansen, Michael Riegler and Pål Halvorsen. 'Predicting Peek Readiness-to-Train of Soccer Players Using Long Short-Term Memory Recurrent Neural Networks'. In: *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*. 2019, pp. 1–6. DOI: 10.1109/CBMI.2019.8877406. URL: https://doi.org/10.1145/2483977.2483982.

[50] *Women's Soccer Attendance Record Smashed By 91,553 Fans As Barcelona Beat Real Madrid*. 2022. URL: https://www.si.com/fannation/soccer/futbol/news/womens-soccer-attendance-record-broken-at-barcelona-vs-real-madrid (visited on 16/05/2022).

Appendix A

Publications and Presentations

The work presented in this thesis has contributed to the following.

A.1 Soccer Athlete Performance Prediction using Time Series Analysis

Authors Nourhan Ragab, Siarhei Kulakou, Cise Midoglu, Pål Halvorsen

Venue Accepted to appear at the NORA Annual Conference 2022[1]

Type Oral presentation

Summary We present our work on predicting soccer players' ability to perform using subjective self-reported wellness parameters including readiness-to-play. We benchmark different time series models as motivated by [20], and further investigate the influence of input and output window size, as well as training prediction models on team data vs.individual data.

[1]**NORA:** Norwegian Artificial Intelligence Research Consortium (NORA) is a Norwegian collaboration between 8 universities, 3 university colleges and 4 research institutes within AI, machine learning and robotics, aiming to strengthen Norwegian research, education and innovation within these fields [30].
NORA Annual Conference: The NORA Annual Conference is an annual event that aims to gather the Norwegian research community within the field of Artificial Intelligence and create a platform where invited speakers and participants can share research, ideas, theories, models and new perspectives, and interact with peers from the field. Knowledge sharing and interaction is at the center of the conference, which the goal is to foster a strong community of researchers and practitioners, while bridging the gap between young researchers, startups and industry [30].

Soccer Athlete Performance Prediction using Time Series Analysis

Nourhan Ragab[a], Siarhei Kulakou[a], Cise Midoglu[a], Pål Halvorsen[a,b]

[a]*Simula Metropolitan Center for Digital Engineering (SimulaMet), Oslo, Norway*
[b]*Oslo Metropolitan University (OsloMet), Oslo, Norway*

Keywords: *athlete training, injury prevention, performance prediction, self-reporting, soccer, time series analysis.*

In this work, we address the problem of predicting soccer players' ability to perform, from subjective self-reported wellness parameters collected using a commercially deployed digital health monitoring system called pmSys [1] (https://www.forzasys.com/pmSys.html). We use 2 years of data from 2 Norwegian female soccer teams, where players have reported daily ratings for their *readiness-to-play* (0-10), *mood*, *stress*, *soreness*, and *fatigue* (0-5). We tackle the so-called "missing data problem" by various methods, including the replacement of all gaps with zeros, filling in average values, as well as removing all gaps and concatenating arrays [2].

We explore various time series models with the goal of predicting *readiness*, employing both a univariate approach (where *readiness* is the only parameter), and a multivariate approach (where *readiness*, *mood*, *stress*, *soreness*, and *fatigue* are the parameters). We provide an experimental comparison of different time series models, such as purely recurrent models (RNN, LSTM [3], GRU, RNNPlus[1], LSTMPLus[1], GRUPlus[1]), models of mixed recursive convolutional types (RNN-FCNPlus, LSTM-FCNPlus, GRU-FCNPlus), ensemble of deep CNN models (InceptionTime), and multivariate versions of the recurrent models (MRNN-FCNPlus, MLSTM-FCNPlus, MGRU-FCNPlus). We use sliding windows of 7, 14, and 21 days as input, and prediction windows of 1 day and 7 days. We also investigate the potential of using data from the whole team for making predictions about individual players. For evaluating the prediction performance, we use the metrics MSE, RMSE, and MAE. For evaluating the system performance, we consider disk space, memory, CPU, GPU, and time usage.

Our case study on athlete monitoring shows that a number of time series analysis models are able to predict performance peaks in most scenarios with a precision and recall of at least 90% in near real-time. Equipped with such insight, coaches and trainers can better plan individual and team training sessions, and perhaps avoid over training and injuries.

References

[1] Kennet Vuong (2014) "PmSys: a monitoring system for sports athlete load, wellness & injury monitoring." *Master Thesis*.
[2] Siarhei Kulakou (2021) "Exploration of time-series models on time series data - A case study on athlete monitoring." *Master Thesis.*
[3] T. Wiik et al. (2019) "Predicting Peek Readiness-to-Train of Soccer Players Using Long Short-Term Memory Recurrent Neural Networks," *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*, pp. 1-6, doi: 10.1109/CBMI.2019.8877406.

[1] These models include an additional feature extractor in the RNN network. This idea comes from the UPSTAGE team (https://www.kaggle.com/songwonho, https://www.kaggle.com/limerobot, and https://www.kaggle.com/jungikhyo), who finished 3rd in Kaggle's Google Brain - Ventilator Pressure Prediction competition, using a Conv1d + Stacked LSTM architecture.

A.2 SoccerMon: A Large-Scale Multivariate Dataset of Soccer Athlete Health, Performance, Training Load, and Position Monitoring

Authors Cise Midoglu, Andreas Kjæreng Winther, Matthias Boeker, Susann Dahl Pettersen, Sigurd Pedersen, Nourhan Ragab, Tomas Kupka, Morten Bredsgaard Randers, Ramesh Jain, Svein Arne Pettersen, Håvard Dagenborg Johansen, Dag Johansen, Michael A. Riegler, Pål Halvorsen

Venue To be submitted to Nature Scientific Data [2]

Type Dataset paper

Summary A key development in sports over the last three decades has been the increased use of scientific methods to improve the preparation for and participation performance in elite competitions. In this context, international sports is undergoing a revolution, fueled by the rapidly increasing availability of athlete quantification data, sensor technology, and advanced analytic software. These recent innovations have enabled a close monitoring of athlete performance across all training sessions and matches, facilitating a much deeper understanding of training methods that benefit athletes and coaches alike, i.e., facilitating a much deeper understanding of training methods and empowering both players and involved personnel to quantify individual and team strengths and weaknesses, give physical performance feedback, psychological status and wellness fluctuations, educate, plan training content in micro- and macro-cycles, avoid overuse injuries and overload, etc.

Overall, data analytics for professional sports has become a prominent area for research, which allows to prevent athlete injuries and optimize performance. However, sports data is often sparse and hard to obtain due to legal restrictions, unwillingness to share, and lack of personnel required for the tedious process of collecting and preparing the data. These constraints make it difficult to develop systems for analysis, especially automated systems, where large datasets are required for learning. We therefore present SoccerMon, the largest dataset available today containing both subjective and objective data collected over two years from 2 different elite female soccer teams in Norway.

The subjective metrics in SoccerMon pertain to wellness, injury, and performance, and are self-reported by players through the use of a mobile application. The objective metrics pertain to location information (GPS) and are collected during training sessions and games through the use of wearable sensors (APEX GPS tracker from STATSports). The dataset has been collected during the 2020 and 2021 seasons in the Norwegian female elite soccer league "Toppserien".

Initial experiments with the large-scale dataset show how various parameters correlate, and demonstrate the potential benefits of AI-based wellness and performance prediction systems. Thus, SoccerMon can play a valuable role in developing better analytic models not only for sports, but also for other fields having subjective reports, position information and/or time-series data in general.

[2] Scientific Data is a peer-reviewed, open-access journal for descriptions of datasets, and research that advances the sharing and reuse of scientific data [21].

A.3 Exploration of Different Time Series Models for Soccer Athlete Performance Prediction

Authors Siarhei Kulakou, Cise Midoglu, Nourhan Ragab, Matthias Boeker, Pål Halvorsen

Venue Submitted to the 8th International Conference on Time Series and Forecasting (ITISE 2022)

Type Journal abstract

Summary We extend our work on predicting soccer players' ability to perform using the subjective self-reported readiness-to-play parameter. We use the full SoccerMon dataset (i.e., 2 years of data from 2 Norwegian female soccer teams, where players have reported daily ratings for their readiness-to-play, mood, stress, general muscle soreness, fatigue, sleep quality, and sleep duration) and the Tsai library. We explore various time series models and provide an experimental comparison of their performance in predicting readiness, with a focus on detecting peaks, employing both a univariate approach, and a multivariate approach. We compare the accuracy of next-day predictions and next-week predictions, and investigate the potential of using models built on data from the whole team for making predictions about individual players, as compared to using models built on the data from the individual player only. We also tackle the so-called "missing data problem" by various methods. Overall, we evaluate both prediction performance and system performance.

Exploration of Different Time Series Models for Soccer Athlete Performance Prediction

Siarhei Kulakou, Cise Midoglu, Nourhan Ragab,
Matthias Boeker, and Pål Halvorsen

Simula Metropolitan Center for Digital Engineering (SimulaMet), Norway

Abstract. Professional sports achievements combine not only the individual physical abilities of athletes but also many modern technologies in areas such as medicine, equipment production, nutrition, and physical and mental health monitoring. Proper diet, rest, and training regimens, as well as continuous monitoring and analysis of wellness and performance metrics can make a big difference for both individual and team sports. Soccer players constantly adhere to a strict nutrition plan, training process, and rest regime so that their body is in the required state at particular moments. The athletes' condition is influenced not only by the amount of consumed and burned calories, or the duration and intensity of the training process, but also by parameters such as the duration and quality of their sleep and their general mental state including mood and stress level. In this context, it is possible to compile a set of parameters describing the general state of the athlete at a certain point of time, collect objective or subjective measurements of these parameters, and try to predict the state/behavior of the athlete's body in the near future.

In this work, we address the problem of predicting soccer players' ability to perform, from subjective self-reported wellness parameters collected using a commercially deployed digital health monitoring system called PmSys. We use 2 years of data from 2 Norwegian female soccer teams, where players have reported daily ratings for their readiness-to-play (0-10), mood, stress, general muscle soreness, fatigue, sleep quality (0-5), and sleep duration (hours). Using the Tsai library, we explore various time series models with the goal of predicting readiness, employing both a univariate approach, and a multivariate approach. We provide an experimental comparison of different time series models, such as purely recurrent models (RNN, LSTM, GRU, RNNPlus, LSTMPLus, GRUPlus), models of mixed recursive convolutional types (RNN-FCNPlus, LSTM-FCNPlus, GRU-FCNPlus), ensemble of deep CNN models (InceptionTime), and multivariate versions of the recurrent models (MRNN-FCNPlus, MLSTM-FCNPlus, MGRU-FCNPlus), in terms of prediction performance, with a focus on detecting peaks. We use windows of 7, 14, and 21 days as input, and prediction windows of 1 day and 7 days, allowing us to compare the accuracy of next-day predictions (which are less relevant in a professional soccer competition context, but interesting for exercise and training purposes) and next-week predictions (which are of interest to teams especially during the match season). We also investigate the potential of using models built on data from the whole team

for making predictions about individual players, as compared to using models built on the data from the individual player only. As a common challenge associated with subjective daily report collection, we tackle the so-called "missing data problem" by various methods, including the replacement of all gaps with zeros, filling in average values, as well as removing all gaps and concatenating arrays. For evaluating the prediction performance, we use the metrics MSE, RMSE, and MAE. For evaluating the system performance, we consider disk space, memory, CPU, GPU, and time usage.

Our case study on athlete monitoring shows that a number of time series analysis models are able to predict readiness peaks in most scenarios with a precision and recall of at least 90% in near real-time. Equipped with such insight, coaches and trainers can better plan individual and team training sessions, and perhaps avoid over training and injuries.

Keywords: athlete training, data imputation, injury prevention, performance prediction, self-reporting, soccer, time series analysis