Recognising Patterns of Play in Rugby Union using Recurrent Neural Networks

Author

Kirk Meyer

Supervisor

Mr. Neil Watson

submitted in partial fulfilment of the requirements for the degree of Bachelor of Commerce (Honours) in Statistical Sciences

in the

Department of Statistical Sciences

University of Cape Town

Submitted:

Declaration of Authorship

copy paste plagiarism declaration here

Abstract

Recognition of patterns of play in the sport of rugby union is a challenging and novel concept. Due to the sequential nature of rugby union, we set out to build a recurrent neural network capable of recognising three pre-determined plays; the exit play, the forward based play and wide-to-wide play. Specifically, we considered a simple Recurrent Neural Network and a LSTM model, along with a non-sequential Random Forest model, as a baseline. The results show that the recurrent neural networks are capable of recognising patterns of play, more so than the random forest model. Indicating that accounting for the sequential nature of rugby union is essential when looking to recognise patterns of play.

***K*eywords** Patterns of Play Recognition, Recurrent Neural Network,Rugby Union

Acknowledgements

Contents

List of Tables

List of Figures

List of Abbrevations

AUC Area Under Curve. 27

CNN Convolutional Neural Network. 7–9

DyCoN Dynamically Controlled Network. 7, 8

FFNN Feed Forward Neural Network. 8, 11, 12

FN False Negative. 15

FP False Positive. 15

FPR False Positive Rate. 16

GRUs Gated Recurrent Units. 32

KFM Kohonen Feature Map. 6, 7

KPI Key Performance Indicator. 5

LSTM Long Short-Term Memory. 2, 3, 9, 12, 14, 20, 25, 27, 29, 32, 33

MSOMS Merge Self-Organising Maps. 8

RNN Recurrent Neural Network. ii, 2, 7–9, 11, 12, 14, 20, 26, 29, 32, 33

ROC Receiver Operating Characteristic. 16

SOMS Self-Organising Maps. 8

TN True Negative. 15

TP True Positive. 15

Chapter 1

Introduction

1.1 Background

A match of rugby union is contested between two teams, each team with 15 players, eight forwards and seven backs. The forwards are generally larger, stronger players, and the backs are more agile and skilled. These players work together, executing specific tactics to invade the opposition territory and score points. A single match of rugby union consists of multiple possessions shared between the two teams. A single possession can be broken down into a sequence of phases, with each phase consisting of multiple actions. Examples of these actions are kicks, passes and tackles amongst many others. Phases generally end with a breakdown, which is when the player with possession of the ball is tackled and a ruck forms. When the ball leaves the ruck, the next phase begins.

Patterns of play are the specific tactics which teams use to invade opposition territory, with the goal of scoring points. A pattern of play may be defined within one phase, several phases or even an entire possession. What defines a pattern of play is a sequence of specific actions in a particular order, with the associated field position and player positions. The patterns of play that we are interested in recognising are the Exit play, the Forward Based play, and the Wide-to-Wide play. Each of these patterns of play is described by truly unique sequences of actions.

There is no previous literature which attempts to recognise patterns of play in the sport of rugby union. The majority of literature focuses on univariate key performance indicators(KPIs) as a basis for performance analysis. Ultimately, this restricts research, making it impossible to account for all the complexities in rugby union. However, if we expand the focus, and use more sophisticated statistical tools, maybe we will be able to account for these complexities. What makes the recognition of patterns of play in rugby union interesting, is the complex sequential nature of the game. Thus, we propose the use of a Recurrent Neural Network (RNN) to recognise these specific patterns of play.

1.2 Research Objective

This project aims to answer two main questions:

1. Can a Recurrent Neural Network be used to recognise specific patterns of play in rugby union?

2. Is it necessary to take into account the sequential nature of rugby union when looking to recognise patterns of play?

To answer these questions, we look to complete three objectives. The first is to build a RNN model, that can intake sequences of data, and recognise the pattern of play. We then seek an optimal model configuration to improve the recognition capabilities of the model. Finally, we build a non-sequential model to compare our sequential model too. This comparison is made in the hope to answer question 2. A Random Forest model will be used as the baseline non-sequential model.

It should be noted that even though patterns of play can occur in single or multiple phases within a possession, for this project, we labelled entire possessions as a play or non-play. Thus, we are looking to recognise which possessions contain a specific pattern play, and which possession do not.

1.3 Outline

This section outlines the remainder of the project:

Chapter 2 focuses on the previous literature that is associated with this research topic. Section 2.1 provides an overview of the statistical research done on the sport of rugby union, discussing how the majority of the literature previously focuses on KPIs, and the recent shift to more modern statistical methods. The discussion then broadens, in Section 2.2 we discuss how supervised and unsupervised methods are used to recognise tactical structures or patterns of play in other sports.

Chapter 3 delves into the methods used in this project(Section 3.1), the data(Section 3.2), and describes the model building process(Section 3.3). First, we present and detail our choice of sequential models, namely, the Simple RNN and Long Short-Term Memory

(LSTM) model in subsections 3.1.1 and 3.1.2. The subsequent subsections in Section 3.1 focus on the remaining statistical methods incorporated into the model building process. The chapter continues, providing a detailed description of the data, and explains every facet regarding the data in this project. The model building process closes the chapter, as we discuss the design of each of the models, and model configurations considered.

Chapter 4 discusses the performance of the models on the test set. We analyse the performance across the different patterns of play, comparing the two sequential models against one another, and the non-sequential random forest models. Section 4.3, reviews what we have learnt from the results.

Chapter 5 suggests considerations for future research work of this nature. These considerations revolve around potential improvements one can make to the research project. Furthermore, it discusses the limitations encountered as well as expansions of the research topic.

Chapter 6 summarises the project, providing an overview of what we have done and the main conclusions that we can draw from it.

Chapter 2

Literature Review

A collection of literature was considered and reviewed from the relevant research areas that the topic encompasses. The review will specifically be on statistical research done on rugby union and tactical structure recognition in other invasion sports.

2.1 Previous Literature in Rugby Union

The area of statistical research on the sport of rugby union is relatively young with literature being published consistently only for the past twenty years or so. This section provides a brief overview of the literature over that period. The focal points being the focus on key performance indicators(KPIs) as a basis for performance analysis in literature, the shortcomings of KPIs and recent advancements in literature.

In its infancy, the focus in the literature has been concentrated simple metrics (KPIs) that discriminate between successful and unsuccessful teams. Performance analysis is an attempt to objectively assess a teams performance through the previously mentioned key performance indicators (KPIs). A KPI is generally a univariate measurement of a teams performance in some facet of the game. Jones et al. [2004] presented a list of team performance indicators which were vetted by elite rugby union coaches. For example, scrum success, time of possession(minutes) and tries scored as a percentage of the total tries scored in a game. Importantly, a teams performance is assessed by comparing which KPIs differ significantly from one another between a winning and losing team. This is done to provide coaches with actionable information to help make tactical decisions.

Despite the popularity of KPIs in literature, there exist plenty of criticisms. One of these criticisms is that research is often done without taking into account contextual factors such as opposition, on-field location and score differential, for example. The suggestion is that these factors heavily influence the indicators, and thus, these indicators do not provide an accurate measurement of a team's performance [Colomer et al., 2020]. However, there have been various attempts to account for these factors in the literature. For example, Wheeler et al. [2013] used the on-field location to contextualise defensive ruck strategies. They found that defending teams were more likely to regain possession via an early counter ruck strategy in the wide attacking channels. An early counter ruck strategy is when the defending team attempt to regain possession by clearing out the opposition protecting the ball in the ruck. In comparison, a jackal orientated strategy was the most successful in the centre channel of the field. A jackal being when a player on the defending team is competing for the ball using their hands after a tackle was made but before the formation of the ruck. These results illustrate the importance of context with the KPIs. It provides potentially an additional layer to the actionable information.

A further criticism is the validity of KPIs. Watson et al. [2017] reviewed the validity of historically statistically significant rugby union team KPIs. An extensive range of KPIs was considered; validation is done using a data-set spanning several competitions and seasons. Ultimately, seeing whether these KPIs continue to be valid differentiators between winning and losing teams. They found that of the 69 KPIs investigated only 12 remained significant across all of the seasons and competitions considered. Furthermore, the majority of those significant KPIs were marginally or even negligibly different in effect size between winning and losing teams. Those significant KPIs that had large effect sizes are considered in the literature to be common sense knowledge in rugby union. Meaning they do not add any new information that could be used by stakeholders. Another shortfall of KPI focused literature presented by Watson et al. [2017] is that the current form of KPIs (univariate measurements) do not capture the complexity in rugby union performance. A potential solution is for more multivariate methods or techniques to be considered to model the complex, multifaceted nature of rugby union.

The literature has moved on from the core focus being on KPIs. Recently more literature has been published were more advance statistical methods have been used. It is not to say that the more advanced the method, the better the analysis, however, these methods do allow for a complete analysis of the complex nature of rugby union.

Croft et al. [2015] used a Kohonen Feature Map (KFM) where the input data a vector of KPIs. This lead to the clustering KPIs and specifically which group of KPIs lead to wins and which attributed to losses in the 2013 New Zealand ITM competition. These groups of KPIs represented potential patterns of play that lead to wins (or losses).

Coughlan et al. [2019] used clustering to identify which patterns of play lead to tries. They also ranked the strength of teams which provided further context, along with the field locations. They found that lineout to maul was the most successful playing pattern. This result illustrates how the advancements of statistical methods used in literature will allow for more contextualised results. More complex or detailed forms of data are now usable, leading to the complexities of the game to be represented in the analysis.

Rugby Union, like many other sports, are Spatio-temporal. However, the sequential aspect is often overlooked. Recently though, Watson et al. [2020] used a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) to predict outcomes of sequences of plays. The considered outcomes were scoring a try, being awarded or conceding a penalty, gaining territory and retaining possession. The performance was compared to a baseline of a Random Forest. The core result is that when field location and sequential data is accounted for the performance of classification improves over the baseline. More importantly, Watson et al. [2020] also integrated their best model for decision support. They used their combined CNN-RNN model, which also accounted more location to predict the outcome probabilities of hypothetical sequences of actions, action descriptions and their field locations. They then presented their predictions, visualised as a “heat map” scenario plot. These scenario plots are an example of how the complex models could be used in a decision support system to provide tactical insight and actionable information for stakeholders.

2.2 Literature on Tactical Structure Recognition

Tactical Structure recognition in other sports has been a more developed area of research compared to rugby union. Perl and Dauscher [2006] expressed that although traditional statistical methods are helpful, they are not sufficient to model the dynamics of a process. To model and understand the dynamics of a process, the time-dependent nature should be considered. This consideration of time-dependency is necessary for the identification of a dynamic process, as a pattern of play. The use of Artificial Neural Networks and other Machine Learning techniques to recognise patterns play has been vastly explored in literature for this very reason. A broad categorisation of these different methods could be unsupervised and supervised methods.

The most popular form of an unsupervised method of tactical structure recognition is Self Organising Maps. More specifically the Dynamically Controlled Network (DyCoN), an extended KFM is popular. Pfeiffer and Perl [2006] analysed the tactical structures of handball and were able to classify different patterns in offensive plays, using the DyCoN model. The focus was to identify the different offence plays, which they were able to do. They found clusters which contained the main offence patterns of play used in handball. Furthermore, this allowed them to identify plays used by different teams. Kempe et al. [2015] investigated the ability of the DyCoN and Merge Self-Organising Maps (MSOMS) to analyse Spatio-temporal data and compared the respective performance of classification of plays in basketball. Each SOMS was fed three pre-selected plays, and the DyCoN had a correct classification rate of 97.5%, while the MSOMS had an 80% accuracy. These results illustrate the ability of a SOMS to find and classify tactical behaviour based on tracking data.

Supervised learning approaches to recognition of tactical structures have become more widely used in since 2013 with the development of various neural networks. They allow for complex, non-linear relationships to be modelled, thus being able to capture the complexities of many sports.

Teich et al. [2016] investigated predicting the success of plays in the NFL. Their focus was on real-time prediction so that coaches can make decisions in-game. They used an extensive set of supervised learning models, ranging from Decision Trees to Neural Networks. Overall, their best performing models were Decision trees and Support Vector Machines using a Radial Basis Function kernel. A criticism of their research was their limited exploration into Neural Networks as they did not attempt to find an optimal network configuration. They also stated that the lengthy running times, defeated the purpose of real-time in-game predictions. However, the different types of neural networks may serve better use in different settings. For example, Mehrasa et al. [2018] used CNNs to recognise in-game events in ice hockey and classify teams in basketball, by taking player trajectories as inputs into the respective models. For the in-game event recognition, they were able to classify two of six events (Pass and Carry) with relative success, with model accuracy being at least 80%. The team classification model aims to recognise the patterns that teams take on during a possession and differentiate teams based on these patterns. They were on average able to classify 95% of the teams when using an entire match of player trajectories as the input.

The dynamics of sports could be explained by a sequential record of events occurring in a game. This opens up the usage of RNNs, alone or in tandem with other neural networks.

The use of RNNs to recognise offence plays in basketball was the focus of Wang and Zemel [2016]. Their data was a pictorial representation of a play, so each player's positioning on the court throughout of play. Using a standard Feed Forward Neural Network (FFNN) and a RNN they were able to classify plays in a fraction of the time a human can.

Baccouche et al. [2010] and Tsunoda et al. [2017] both used RNNs, more specifically a Long Short-Term Memory (LSTM) network to classify actions in football by translating video data into sequential data. Baccouche et al. [2010] used sequential feature descriptions of the video data, while Tsunoda et al. [2017] improved their methodology using a CNN to extract feature's present in the video data.

The uses of RNNs in sports stretches far beyond tactical structure recognition, for example, Shah and Romijnders [2016] predicted whether or not a three-point shot was made based on the sequential trajectory of the basketball.

The literature highlights an essential point that the nature of sport is complex. Thus, as the statistical modelling methods advance more complex data can be analysed capturing the complexities of the game. While the majority of literature uses frequency data or unsupervised learning methods to analyse patterns of play in rugby, this project proposes the use of a supervised learning approach to recognition of patterns of play in rugby union.

Chapter 3

Methodology

3.1 Methods

When undertaking research, the selection of methods used is critical. This chapter explains the choice of specific models and the underlying theory behind them. We briefly discuss the limitations of these modelling techniques. Moreover, we present the measures of performance used and touch upon the optimisation process during the learning of the neural networks.

3.1.1 Simple Recurrent Neural Network

Considering that the data is sequential, using a Recurrent Neural Network (RNN) is a natural starting point. It has cyclical connections which differentiate it from standard Feed Forward Neural Network (FFNN). The implication of this being that RNN's can map previous inputs to outputs. It is often described as a FFNN, with internal memory. This makes RNNs an attractive method, as they can learn what information in a sequence is important, and are able to accept different forms of data as inputs, making RNNs useful for recognising sequential patterns [Graves, 2012].

The first RNN model used in the project is best described as a simple or vanilla RNN. There are many RNN architectures, for example, one-to-many, many-to-one, and many-to-many which determine the number of units in the input and output layers. A RNN processes sequences of data points, iteratively going through the elements of the sequence while recording what was seen so far [Chollet and Allaire, 2018]. Thus, a sequence is considered as a single input, into a RNN. The hidden layers receive current external inputs and hidden layer activations from previous time-steps of a sequence [Graves, 2012]. This process is continued until the final element in the sequence is reached, after which an output is produced. This, in fact, describes the forward pass of inputs through the RNN.

Similarly, to FFNN if one is using a gradient-based learning method, backpropagation would be used to evaluate the gradients used to update the weights and biases, throughout the network such that the cost function is minimised. Named backpropagation, since gradients are evaluated starting at the cost function and works backwards throughout the network, evaluating gradients for each parameter in each layer [Rumelhart et al., 1986].

The major limitation with Simple RNNs, is that they struggle to recall information from earlier on in a sequence, due to the vanishing gradient problem [Hochreiter, 1998]. The design of the Simple RNN requires the algorithm to propagate through each time step to evaluate all of the gradients. Furthermore, there is only one common weight that connects the hidden layers at each time step with each other. If this gradient is small, preceding gradients would seemingly vanish, resulting in "forgetting" what occurred earlier on in the sequence.

3.1.2 Long Short-Term Memory

The problem of vanishing gradient has limited the use of Simple RNN's. There have been many attempts at resolving the issue, for example, using non-gradient based learning methods. However, the most popular solution has been the Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997]. The LSTM is similar to the Simple RNN but differ in the design of their hidden layers. The hidden layers of a LSTM are in literature commonly referred to as memory blocks, illustrated in Figure 3.1.

The architecture of these blocks is what solves the issue of vanishing gradient. For this project, we considered the simple design of the LSTM. In each block, there is a single memory cell(cell state) and three multiplicative units, namely, the input, output and forget gates. These three gates control the flow of information in and out of the cell state. Firstly, the forget gate determines if information in the previous cell state should be kept or forgotten, given the new input that has entered the network. The input gate updates the current cell state by determining if the current input should be taken into account in the current cell state. The output gate determines which information that is present in the current cell state is passed into the network at the next time step.

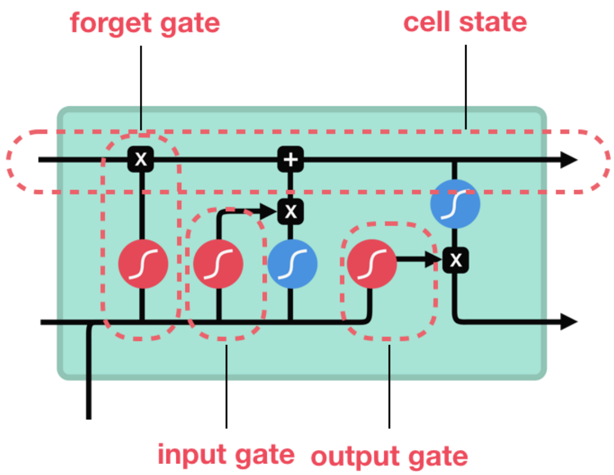

Figure 3.1: LSTM Memory Block

This architecture, along with the adapted backpropagation algorithm, resolves the issue of the vanishing gradient [Hochreiter and Schmidhuber, 1997]. The effect is that the gradients will less likely vanish in a short period of time. Furthermore, the presence of the forget gates produces more stable solutions as the memory cells can reset when needed [Gers et al., 1999].

3.1.3 Learning Algorithm

The learning problem is finding the optimal set of parameters which minimises the loss function without overfitting. The loss function is a function of the parameters of the neural network, which measures how well the model fits the data. Thus, the learning of the neural network aims to find suitable adjustments to the set parameters. At each iteration, the algorithm determines how the parameters are adjusted such that the loss function is minimised. This is repeated until some convergence criterion is met. However, to prevent overfitting, we regularise the parameters by considering a penalised objective function. This can be achieved by adding a penalisation term on to the loss function, as seen here:

$$C = \underbrace{-\sum_{i=1}^{N}\left(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right)}_{\text{Binary Cross-Entropy Loss Function}} + \overbrace{\lambda \boldsymbol{W}^T \boldsymbol{W}}^{\text{L2 Norm Penalty Term}}$$

where N is the number observations in data, $\boldsymbol{W}$ is the weight vector, which consists of all the weights present in the neural network, and λ is the regularisation parameter which determines the degree of regularisation. The goal is then to find to the optimal set of parameters which minimises penalised objective function.

How the adjustments are made to the parameters of the model, describes the learning algorithm. There are a wide variety of classes of learning algorithms such as the gradient-based learning algorithms. For this project, we used the Adam optimiser, which is a gradient-based learning algorithm. The algorithm, based on estimates of the first and second-order moments of the parameters, takes the best from two popular learning algorithms, namely, AdaGrad [Duchi et al., 2011] and PMSProp [Tieleman and Hinton, 2012].

The updating equation of Adam, is defined as [Kingma and Ba, 2014]:

$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \alpha_t \cdot \frac{\boldsymbol{m}_t}{\sqrt{\boldsymbol{v}_t + \epsilon}},$$

where,

$$\alpha_t = \alpha \cdot \sqrt{1-\beta_2^t} \Big/ (1-\beta_1^t).$$

The parameter vector at time t is given by $\boldsymbol{\theta}_t$ and the low-order moment estimates are calculated as,

$$\boldsymbol{m}_t \leftarrow \beta_1 \cdot \boldsymbol{m}_{t-1} + (1-\beta_1) \cdot \boldsymbol{g}_t \quad \text{and} \quad \boldsymbol{v}_t \leftarrow \beta_2 \cdot \boldsymbol{v}_{t-1} + (1-\beta_2) \cdot \boldsymbol{g}_t^2,$$

where $\boldsymbol{g}_t$ is the gradient evaluated at t and $\beta_1, \beta_2 \in [0,1)$ are the exponential decay rates for the moment estimates. Kingma and Ba [2014] suggests that $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\alpha = 0.001$ and $\epsilon = 10^{-8}$. The gradients are calculated using backpropagation for both RNN [Williams and Zipser, 1995], and LSTM [Hochreiter and Schmidhuber, 1997].

3.1.4 Random Forest

A Random Forest [Breiman, 2001] is a robust supervised learning algorithm, which takes bagging or bootstrap aggregating of classification trees, and improves upon it.

Classification trees are used to make predictions on categorical or binary responses. It makes use of the recursive binary splitting algorithm, which splits the predictor space into high dimensional regions until some stopping criterion is reached [James et al., 2013]. The splits that are made in each step cause the largest reduction in some measure of deviation or variability within the model.

The issue with simple classification trees is that they are plagued with high sample variability.

Meaning, if we were to fit a tree with new samples from the same population, the results may differ greatly [James et al., 2013]. Thus, bagging was introduced to reduce this sample variability. The data set would be made into B bootstrap samples, and B trees would be built from these samples [Breiman, 1996]. In the case of classification trees, the most common of B response is selected. The problem with this method is that if there exists a dominant predictor, the predictor space will be split in similar ways. This causes the B trees to be highly correlated.

Random Forest solved this by decorrelating the trees. At each split of the predictor space, a random sample of $m < p$ predictors are considered, where p is the total number of predictors [Hastie et al., 2009]. This ensures that the resulting trees are decorrelated.

3.1.5 Evaluation of Performance

A variety of classification-based performance metrics were selected to assess individual model performance. Moreover, these metrics are used as a measure of comparison against other models. All of these metrics are derived from the results in a confusion matrix, Table 3.1.

		True	
		Non-Play	Play
Predicted	Non-Play	TN	FN
	Play	FP	TP

Table 3.1: Confusion Matrix

True Negative (TN) represents the number of possessions that do not contain a specific play, and were predicted as such; False Negative (FN) represents the number of possessions that do contain a specific play but were predicted not to contain the specific play. False Positive (FP) and True Positive (TP) are described in the same manner. The metrics which are derived from these results are:

$$
\begin{aligned}
&\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN + FN + FP +TP}},\\
&\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}},\\
&\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}},\\
&\text{False Positive Rate (FPR)} = 1 - \text{Specificity},\\
&\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},\\
&\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}},
\end{aligned}
$$

Accuracy measures the proportion of total observations which were correctly classified. Sensitivity measures the proportions of plays that were correctly classified. Since the objective of this project is to recognise patterns of play, sensitivity is closely analysed. Specificity measures the proportion of non-plays that were classified correctly. The F1 Score is a metric which provides a balanced measure of precision and sensitivity. Precision being a measure of how exact the predictions are, as it is the proportion of predicted plays that were correct. The final measure of performance is the area under the Receiver Operating Characteristic (ROC) curve (AUC), which is found by plotting sensitivity against FPR for a range of probability thresholds. This measure is representative of the predictive power of the model.

3.2 Data

The data is spatiotemporal, and thus provides all the critical information needed to build a model to recognise patterns of play. Data pre-processing forms an integral part of this project, where the labelling of the data is crucial to facilitate the use of supervised learning methods. Thus, this section offers a detailed description of the data, explains the pre-processing of the data, and discusses the three patterns of play. Moreover, we touch on data handling and the embedding of categorical data.

3.2.1 Data Description

The large dataset consists of game data from the Super Rugby 2015 season. Each observation has multiple features, each of which is described in the table 3.2:

Feature Name	**Feature Description**
id	Identification Number of Observation
act	The Action Group Qualifier
act_type	Action Type Qualifier
act_res	Result of Action Qualifier Description
x_crd and y_crd	X and Y coordinate at start of an observation
x_end and y_end	X and Y coordinate at end of an observation
position	Player Position
pos_num	Possession number in match

Table 3.2: Feature Name and Descriptions

There are 29 categories of actions group qualifiers recorded in the data, for example, a "kick", a "pass", or a "tackle". These actions are described in further detail through the Action Qualifier variables. There are 475 unique action type descriptors, for example, a kick may be any one of the following: "box kick", "bomb" or "territory" each of which adding detail to the action group qualifier. Each action leads to an action result, which is also described by one of the unique 475 action type descriptors. For this project, we also consider the player position, the x and y field locations for each action, and the x and y field end locations for each action result. The x and y coordinates are in terms of the action performing team, with x = 0 representing the action performing team's try-line, x = 100 the opposition's try-line, y = 0 the left-hand touchline and y = 68 the right-hand touchline of the action performing team. All the features are categorical variables except for the coordinate features, which are continuous variables.

3.2.2 Data Pre-Processing

The structure of the raw data is such that it cannot be considered as an input into the models. The first step is to remove any unused features from the raw data, resulting in the features presented in Table 3.2. After which the sequences of observations are divided into possessions. Each possession is then labelled as a play if it contains one of the three patterns of play, or a non-play if it does not contain the pattern of play of interest. Action Group, Action Type, Action Result, the two pairs of x and y coordinates, and the player position features are then extracted from the data. The key features are the input streams into the neural network models. We are left with multiple sequences for each key feature, one sequences representative of a single possession. Each sequence was 100 elements in length, pre-padded with zeros if necessary ensuring that sequence length is fixed. The two coordinate features are also normalised, by the formula:

$$z^* = \frac{z - \min(z)}{\max(z)},$$

where z^* is the normalised coordinate, and $0 \le z^* \le 1$. This is the final step the of the data preparation.

Possession								
Action	RK*	COL	C	TK*	TK*	RCK	P	K
Action Type	50m*	ReC	RR	LT*	ET*	0	CP	T
Action Result	ROC*	SUC	TN	COM*	COM*	W	0	CF
x - coordinate	50*	10	10	90*	90*	27	27	13
y - coordinate	34*	8	8	60*	60*	15	15	32
x - end	89*	0	0	0*	0*	0	0	69
y - end	61*	0	0	0*	0*	0	0	22
Position	FH*	FB	FB	PRP*	LK*	NONE	SH	FH

Table 3.3: Example of the eight input streams for a possession

Table 3.3 illustrates data before being embedded and normalised. Each vector here represents one sample of each feature. The elements that have an asterisk relate to the team that does not have possession of the ball throughout the example. The possession started with a restart kick (RK*) from 50-metre line (50m*) and resulted in it being collected by the opposition (ROC*). The full back (FB) on the opposition team collected (COL) the ball successfully (SUC), and carried (C) it into contact resulting in a neutral tackle (TN). The next two elements detail that tackle (TK*), how the prop (PRP*) and the lock (LK) both tackled the full back (FB) with a line tackle (LT*) and edge tackle (ET*) respectively. This resulted in a ruck (RCK), which was won (W) by the team who previously had possession of the ball. The scrumhalf (SH) then passes (P) the ball from the base of the ruck to the flyhalf (FH) who kicks (K), looking for territory (T), resulting in the opposition catching the ball out the air (CF).

Additionally, COM* refers to a completed tackle by the opposition, RR refers to restart return which is a type of carry off a restart kick, and CP refers to complete pass. Moreover, each action in the sequence has a corresponding (x,y) coordinate, and (x,y) end coordinate. Many of the (x,y) end coordinates are zero; these coordinates are only non-zero after an on-field action results in a large gain in territory, generally occurring through a kick. Ultimately, this displays how rich the data is with information, capturing the complex nature of rugby union.

3.2.3 Label Description

Each pattern of play has a unique sequence of actions that defines it. Thus, when labelling the patterns of play during the pre-processing of the data, it is vital to understand which actions constitute these three plays. To provide insight into the process of labelling the plays, we will describe each play.

The Exit Play is used when the team in possession of the ball are under pressure by the opposition, deep within their own territory. The purpose an Exit Play is to relieve the pressure, by kicking the ball down the field. This moves the action of the match away from their own try-line into more neutral, or opposition territory. The defining sequence of events would be that the ball gets passed from the ruck, generally by the scrumhalf to the team's designated kicker, who kicks the ball down the field, or out of bounds. However, there exist small variations of the Exit Play, for example, the scrumhalf can kick the ball directly from the ruck, or the starting point can be a pass from scrum or line-out instead of a ruck. The key features within the sequence are the kicking action and an x-coordinate of less the 22m. Figure **??** illustrates the Exit Play.

The Forward Based play is a tactical structure that a team takes on to breakdown the opposition's defence. It is best described as forwards carrying the ball into contact phase after phase looking to gain territory, and break the opposition's defensive structure. It is generally used when the attack has stagnated, and the opposition's defence is holding up. The key sequence is a pass from the scrumhalf to a forward who carries the ball into contact, gets tackled after which a ruck forms and this process is repeated several times. The key features within the sequence are multiple carries by a forward, in quick succession. Figure **??** illustrates the Forward Play.

The Wide-to-Wide Play is an attacking structure where the team with possession passes the ball across the field, into the outer channels. The aim is to get the ball in the hands of quicker, more agile backline players, who can create scoring opportunities. The essential characteristic of this play is a sequence of 3 to 4 passes in a row. Generally, we also see a large change in y-axis value. Figure **??** illustrates the Wide-to-Wide Play.

3.2.4 Data Handling

The data was randomly split into training, validation, and test sets with a 70:15:15 ratio. Furthermore, there was class imbalance present in data, where occurrences of plays were the rarer event. We used upsampling as the method of rebalancing the data. However, we only upsampled on the training set, while the validation and test sets remained imbalanced.

The model training was done on both the imbalanced and rebalanced training set to explore how the rebalancing would impact the model performance.

3.2.5 Embedding

Generally, categorical data is represented with one-hot encoding. One hot encoding is a vector representation of a categorical data point, where there are as many features(columns) as the unique number of categories for that variable. A 1 is assigned for the feature representing that data points category and rest of the features are marked 0. There are many issues with this representation; for example, they are computationally expensive, or when there are categories with many unique features, it generates sparse data.

The embedding of categorical variables resolves some of the limitations of one-hot encoding. Guo and Berkhahn [2016] describes embedding is the mapping categorical variables into a vector of n-dimensions, where n < the unique number of categories for that variable. The reduction in dimension improves the speed of training and the performance of neural networks. The optimal mapping is also learned during the training process of a supervised learning method [Guo and Berkhahn, 2016].

3.3 Model Building

The Simple RNN and the LSTM serve as the focal point in the investigation into recognising patterns of play. Independent models are built for each pattern play identified in Section 3.2.3, with 2 model variations for each, one trained on the imbalanced data and another on the rebalanced data. Moreover, we considered a basic Random Forest as a non-sequential baseline to compare the two sequential models too. The purpose of this section is to discuss each of our models' general architecture and everything that it encapsulates. We briefly discuss our hyperparameter tuning experiment. Lastly, all the models were built in `R` [R Core Team, 2020].

3.3.1 Sequential Models

The general architecture of RNN and LSTM models are similar, with the only difference being the type of recurrent layer within the model. The architecture of the sequential models is illustrated in Figure 3.2. It is a multi-input model, with eight input streams where four are categorical (3 Action-related features and Player Position) and four numeric input streams (the pair of 2 (x,y) coordinate features). Each of the categorical inputs is fed into their own embedding layer, where they are converted to vector representation with reduced dimension as explained in Section 3.2.5. More specifically, the Action and Position both having embedding dimensions of 5, while Action Type and Action Result both have embedding dimensions of 25. This greatly reduces the dimension of the inputs into the recurrent layer. The outputs from the recurrent layers are concatenated and passed into a fully connected layer, with a ReLu activation function, before entering the final output layer. The output layer makes use of the sigmoid activation function. Moreover, we used binary cross-entropy as the loss function, and the Adam optimiser with the parameters suggested by Kingma and Ba [2014], stated in section 3.1.3. Furthermore, we used the L2-norm to penalise the weights of the networks, thus avoiding overfitting. The models were fitted using the `keras` package [Allaire and Chollet, 2020].

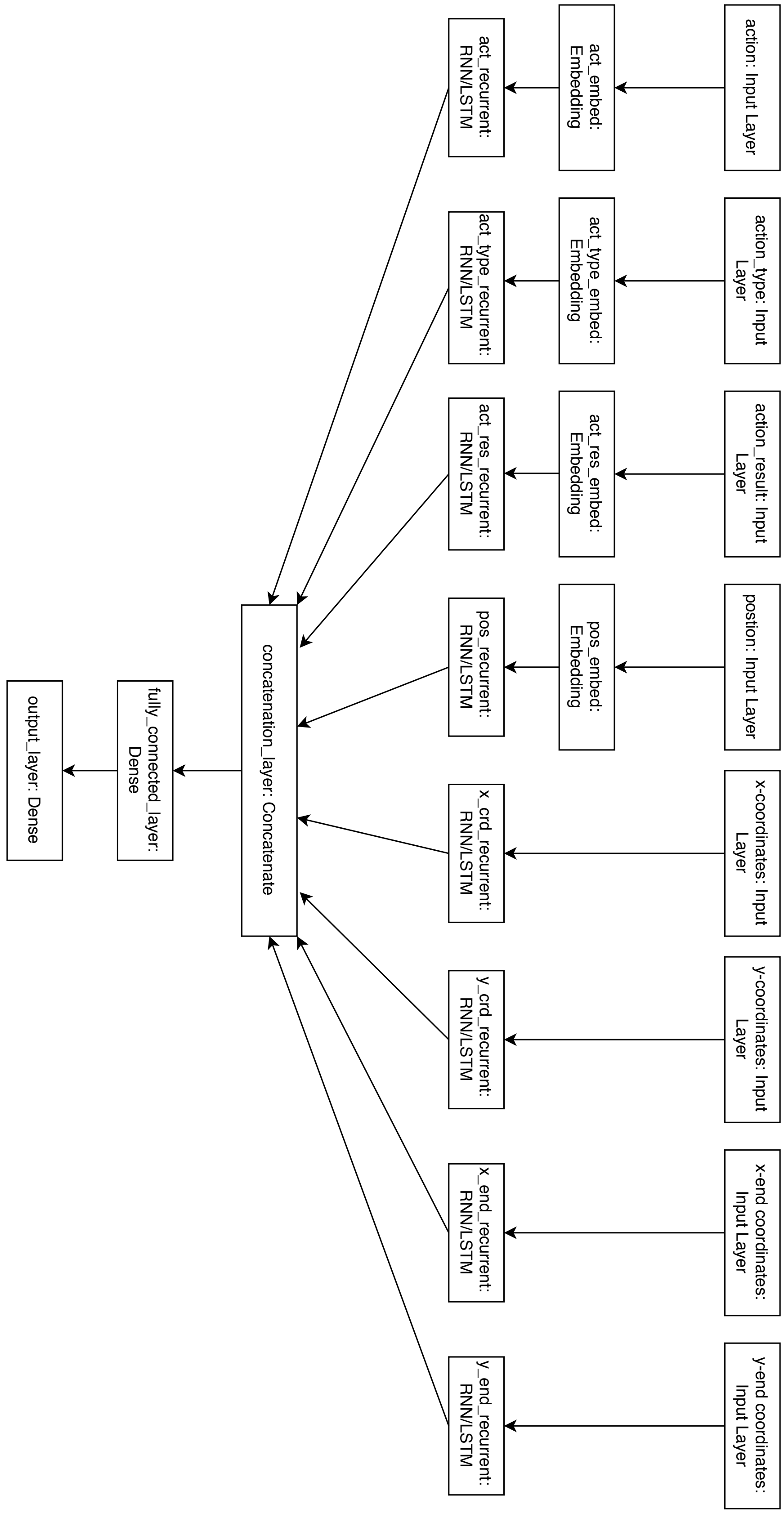

Figure 3.2: Model architecture of the Simple RNN

3.3.2 Hyperparameter Tuning

To find the optimal model, we had various configurations based on different combinations of minibatch size, and the value of the regularisation parameter, λ. We considered 3 minibatch sizes (32, 128, 256) and 6 different λ values ($\lambda = 0, 0.001, 0.01, 0.1, 1, 10$). This resulted in 18 different configurations for each sequential model built, thus across the three patterns of play there is a total of 216 models. The validation accuracy was used as the measure to determine which configuration was best and will be used to assess the performance of the model on the test set.

3.3.3 Random Forest

Considering that the Random Forest is a non-sequential model, the design was such that each time step in an input sequence of a specific feature had its own variable. We considered two variations of a basic Random Forest model, one with only the Action input sequence, and another with Action and Player positions input sequences. Since the goal of the project was the building, and evaluation of the sequential models, we did not explore any tuning of the parameters of the Random Forest model. Thus, the Random Forest model was fitted using the `randomForest` package [Liaw and Wiener, 2002], with the default parameters of 500 trees, and the number of variables considered at each split equalling the sum of the length of the input sequences.

Chapter 4

Results

In this chapter, we first discuss the results obtained from the hyperparameter experiment. From this experiment, we select the model configuration for each network type across each pattern of play, based on the validation set performance. After that, we investigate the performance of these models on test data, using different metrics. We then compare the performance of the sequential models to that of the random forest models.

4.1 Hyperparameter Tuning Results

As stated in Section **??**, there was a total of 216 different model configurations to be evaluated. We select the configuration that results in the highest validation set accuracy. This is done for each type of sequential model, resulting in 12 final models for further investigation.

Figure 4.1 illustrates the validation accuracy for each LSTM model configuration, attempting to recognise the forward based play. The model configuration that resulted in the highest validation accuracy of 93.64% had batch size 128 and a regularisation parameter, λ, equalling 0.01. This model configuration is then used as the final LSTM model trained on imbalanced data when attempting to recognise the forward based pattern of play. This describes the process of selecting the best model configuration to conduct further analyses.

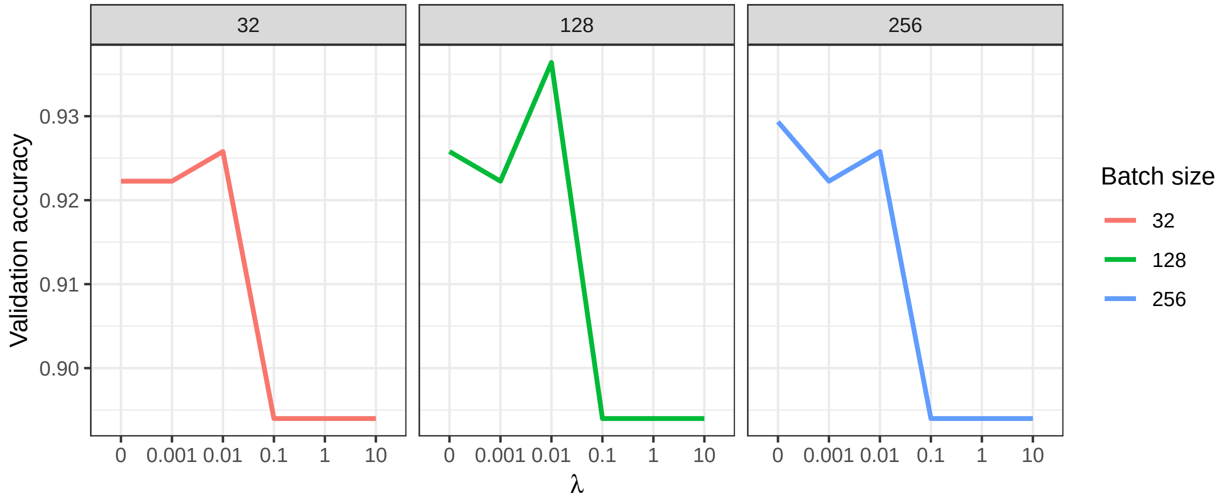

Figure 4.1: Validation Accuracy of LSTM model recognising the Forward Based Play (trained on imbalanced data)

However, there is a situation where selecting the best model configuration is not as straightforward. Figure 4.2 illustrates a situation when two model configurations share the highest validation set accuracy of 95.61%. One with mini-batch size 128 and $\lambda = 0.1$, and another with mini-batch size 256 and $\lambda = 0.01$. In total, there were five situations like this. When in this situation, we selected the model with the larger mini-batch size, as these models update the weights less often making them computationally more efficient while achieving the same validation set accuracy. Therefore, the model with mini-batch size 256 and $\lambda = 0.01$, was the Simple RNN model configuration used, when attempting to recognising the exit play. The remainder of these validation accuracies against model configurations plots are found in the appendix.

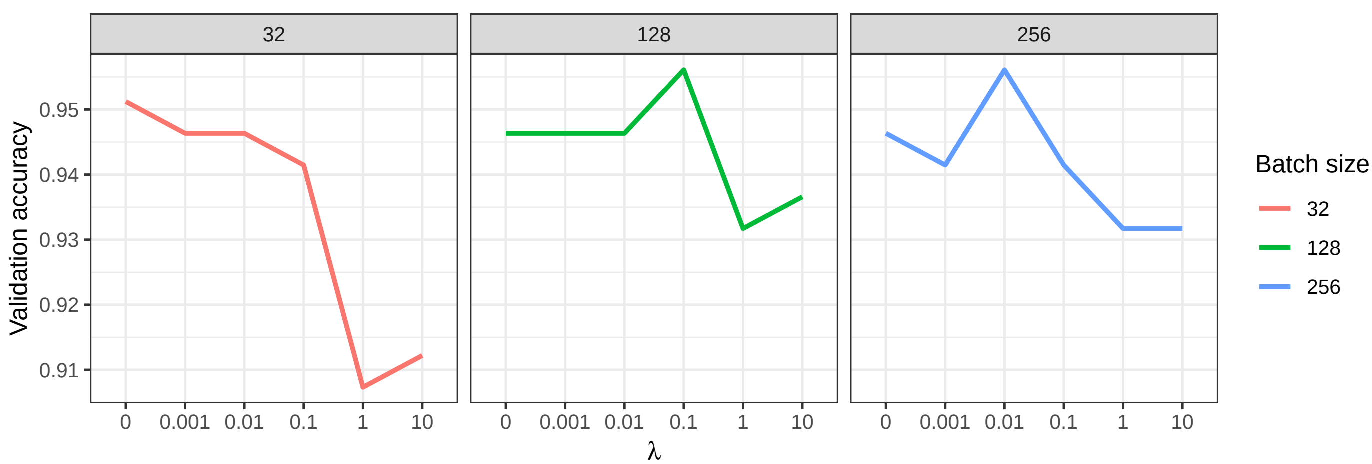

Figure 4.2: Validation Accuracy of Simple RNN model recognising the Exit Play (trained on imbalanced data)

4.2 Model Performance

Table 4.1 presents the performance of each final model. The performance metrics are found with a classification threshold of 0.5 for all the models considered. In this section, we discuss the results, looking to find the best sequential model for each pattern of play. After that, we compare the sequential models to its non-sequential counterpart.

4.2.1 Sequential Model Performance

When looking to recognise the exit play, the accuracy of the four sequential models are incredibly similar, but ultimately the LSTM had the highest accuracy with 91.26%. However, the sensitivity metric for each exit play model is disappointing, ranging between 58.33% and 66.67%. For example, the LSTM model has a sensitivity of 62.50%, indicating that it correctly recognises 62.50% of exit plays. It does have a specificity of 95.05%, meaning that it recognises the non-exit plays well. These results may lead to the assumption that these models are quite average at recognises exit plays and better at recognising what is not an exit play. However, the AUC of these models range between 88.12% and 91.05%, signifying that they have high predictive power. When looking to recognise exit plays, all these models are similar across all metrics, but the LSTM trained on imbalanced data is marginally better than the others.

Shifting to the forward based play, all of these models had high accuracy values, but the LSTM trained on imbalanced data had the highest with 95.05%. Interestingly, the LSTM trained on rebalanced data had the highest sensitivity of all the models by quite a margin, achieving a value of 80%, indicating that it does a decent job at classifying plays correctly. The LSTM trained on imbalanced data had the highest F1 Score with 74.07% and an AUC value of 97.01%. When considering all the metrics, the LSTM trained on imbalanced data seems to just nudge ahead of the others as the best model.

Lastly, we consider the Wide-to-Wide play, where all the models performed well across all metrics. Every model had an accuracy value above 96%, each with high sensitivity and specificity values. This indicates that each model can classify the Wide-to-Wide Play successfully. Overall, the most well-rounded model seems to be the LSTM trained on imbalanced data, with the maximum predictive power (AUC)(99.39%), F1 Score(93.55%) and accuracy(98.35%). However, the difference is marginal for most metrics.

4.2.2 Random Forest Performance

The performance of the random forest models is as expected when considering that these models do not take into account the sequential nature of the data. All 6 of the random forest models have high accuracy values, however, they do have relatively low sensitivity values, with the highest being 67.50% from the Action Position Model attempting to recognise the Wide-to-Wide Play. When considering all the performance metrics, the Action Position Models across all three patterns of play are better than the Action only models. This suggests that if the random forest considered more features of the data, it might have performed better.

However, the purpose of these models, was as a baseline to compare the sequential models too. As the results suggest, the best sequential model for each play outperforms the best random forest model for each play. Even if the models have similar accuracy values, for example, the exit play models (i.e. Exit LSTM and Exit RF Action Position), what differentiates the models are the sensitivity values. Without accounting for the sequential nature of the data, the task of recognising these patterns of play becomes tougher. Ultimately, the random forest models are unable to learn the sequences of key features required to recognise these patterns of play.

4.3 Results Summary

The results suggest that the sequential models are capable of recognising these patterns of play. In particular, the Wide-to-Wide play seems to be easily recognisable, as all four sequential models have high sensitivity values (the lowest being 87.1%). Even the random forest models recognise the Wide-to-Wide Plays better than the other two patterns of plays. This may be due to the design of the Wide-to-Wide play, the key feature being multiple passes, one after the other. This sequence may be less nuanced and more straightforward for the models to "remember" compared to the others.

The sequential models' performance when recognising the Exit and Forward based play at first glance is average at best. However, the data set is heavily imbalanced; only 12.98% of all the data is an occurrence of a play. Taking this into account, along with the high predictive power of these models, a classification threshold of 0.5 may not be the most optimal. There may exist a threshold that optimises the sensitivity, thus, improving the recognition of the models when attempting to recognise the exit and forward based plays. Further investigation into this is required.

When comparing the two types of sequential models, the LSTM models do seem to be marginally better than the Simple RNN. The Simple RNN only betters its LSTM counterpart's (i.e. imbalanced or rebalanced training) performance in 5 out of a possible 30 comparisons of metrics. However, these differences are marginal, indicating that the sequential models are comparable. Furthermore, it is interesting to note that the models that were trained on the rebalanced data do not seem to perform better than the those trained on the imbalanced data.

Model	λ	Mini batch	Accuracy	Sensitivity	Specificity	F1 score	AUC
Exit Simple RNN	0.01	256	0.8981	0.5833	0.9396	0.5714	0.9105
Exit LSTM	0.001	128	0.9126	0.6250	0.9505	0.6250	0.8871
Exit Simple RNN (Up sampled)	1	256	0.8883	0.6667	0.9176	0.5818	0.8812
Exit LSTM (Up sampled)	0	32	0.8981	0.6250	0.9341	0.5882	0.9059
Forward Simple RNN	0	256	0.9399	0.5667	0.9842	0.6667	0.9655
Forward LSTM	0.01	128	0.9505	0.6667	0.9842	0.7407	0.9701
Forward Simple RNN (Up sampled)	0.001	256	0.9222	0.6000	0.9605	0.6206	0.9382
Forward LSTM (Up sampled)	0.01	32	0.9011	0.8000	0.9130	0.6316	0.9569
Wide-to-Wide Simple RNN	0	32	0.9711	0.9677	0.9716	0.8955	0.9914
Wide-to-Wide LSTM	0	32	0.9835	0.9355	0.9905	0.9355	0.9939
Wide-to-Wide Simple RNN (Up sampled)	0	32	0.9669	0.8710	0.9810	0.8710	0.9752
Wide-to-Wide LSTM (Up sampled)	0.1	128	0.9835	0.9032	0.9953	0.9333	0.9891
Exit RF Action			0.8783	0.4400	0.9067	0.3056	0.7879
Exit RF Action Position			0.8881	0.5200	0.9119	0.3611	0.8165
Forward RF Action			0.8940	0.5000	0.9514	0.5455	0.9192
Forward RF Action Position			0.8975	0.5135	0.9553	0.5672	0.9293
Wide-to-Wide RF Action			0.8965	0.6111	0.9324	0.5690	0.9186
Wide-to-Wide RF Action Position			0.9034	0.675	0.9400	0.6585	0.9147

Table 4.1: Model Performance

Chapter 5

Discussion

This chapter discusses the future considerations one could make when researching a topic of this nature. We also mention the limitations we encountered during the completion of this project and present possible topics to research in the future.

5.1 Future Considerations

A more thorough analysis of model selection is something to consider. This can be achieved through various avenues, for example, performing a variable/feature selection. It would be interesting to see which features are essential in recognising the different patterns of play. It may lead to a reduction of parameters within the model, potentially reducing computation times, or the network may lose the ability to recognise specific patterns of plays. Nonetheless, feature selection sparks an interesting discussion.

Hyperparameter tuning is another tool to analyse model selection. Expanding the list of current hyperparameters would be an exciting addition to a future project. Investigating how many recurrent layers to include, or considering a wider range of regularisation parameters could add a layer of rigour to one's analysis.

The method of evaluating performance is also a point of consideration. k-fold cross-validation is an example of such a method. This technique randomly splits the training set into k folds of the roughly the same size. It then treats 1 of these folds as a validation set and then fits on remaining k-1 folds. This process is repeated k times, each time with different fold becoming the validation set [James et al., 2013]. This can be useful in two ways, the first being that one can estimate the test performance metrics by taking the average of the k validation performance metrics. The second use is that one can use this average validation error value to select the best model configuration when doing hyperparameter tuning, then evaluate that model on a test set.

An investigation into the optimal classification threshold should be considered. Since the majority of the sequential models had high predictive power, the classification threshold, which balances the sensitivity and specificity best, should be found. This would provide a fairer reflection of the performance of the models, particularly those recognising the exit and forward based plays.

5.2 Limitations

A major limiting factor was the time consumed by the data labelling. This resulted in us only being able to label a relatively small amount of possessions as plays or no-plays. In total there were 200 occurrences of each play and with 1181 non-exit plays, 1484 non-forward based plays, and 1203 non-wide-to-wide plays. The models were exposed to a small amount of data, and it would have been interesting to see how the models would have performed had there been more data.

Computing resources also limited the project. These models were computationally expensive to run even with the smaller amount of data. This made the considera a broader range of hyperparameters when looking for the best model configuration.

5.3 Future Research

This project illustrates that specific patterns of plays can be recognised by Recurrent Neural Networks. Future research could use these methods to investigate more rugby specific questions. For example, once the best model is found, one can investigate which patterns of play lead to more points scored, or more wins.

Research into the performance of other recurrent neural networks would be interesting, such as Gated Recurrent Units (GRUs) and Bidirectional RNN, just as a comparison to the Simple RNN and LSTM. Additionally, building a multi-class model that can classify all three plays should be considered. It would be interesting to see the performance of single multi-class model compared to the individual play specific model.

Chapter 6

Conclusion

This project aimed to determine whether or not specific patterns of play can be recognised by a Recurrent Neural Network (RNN). The foundation of this research question is built on the sequential nature of patterns of play. Watson et al. [2020] states that there is no action performed within an invasion sport, without the consideration of the actions that came before it. Thus, the use of a RNN model became an obvious one. Specifically, our models of choice were the Simple RNN and the LSTM.

The patterns of play that we set out to recognise were the Exit Play, the Forward Based Play, and the Wide-to-Wide play. Each play selected being intentionally different from one another, to highlight the unique and complex tactical structures within rugby union. To describe the plays, we turned to the features, which encompassed not only the actions done on the field but also who performed the actions and where they occurred. However, due to the time-consuming nature of the data wrangling, the number of observations was underwhelming. Additionally, the data set was extremely imbalanced, with only 12.98% of the observations being the positive label (i.e. a play). To counter this, we upsampled the training data, in an attempt to increase the number of positive labelled observations for the models to train on.

Once the data was prepared, we built our sequential models and evaluated their performance. Moreover, we considered a non-sequential model, the Random Forest, to investigate if the consideration of the sequential nature as was as crucial as believed. The results indicate that accounting for the sequential nature is essential, as every sequential model outperformed its non-sequential counterpart at recognising the play under consideration. Focusing on the sequential models, though the performance across each play was not spectacular, the models had great success recognising the Wide-to-Wide play. However, the models had less success when predicting the other two plays. Interestingly, these models did have high predictive power across all three plays, indicating that the current classification scheme is not optimal. Finding the optimal classification threshold would lead to improved performance when recognising the Exit and Forward based plays.

Ultimately, the results do suggest that recurrent neural networks are capable of recognising patterns of play in rugby union. The random forest models' performance highlights the importance of taking into account the sequential nature of the play. When the model is unable to treat the data as a sequence, it cannot accurately recognise the plays. The architecture and design of the RNN models allow for the learning and remembering the key sequences of a play. Thus, we have shown that Recurrent Neural Networks can be used to recognise specific patterns of play in rugby union.

Bibliography

JJ Allaire and François Chollet. *keras: R Interface to 'Keras'*, 2020. URL https://CRAN.R-project.org/package=keras. R package version 2.3.0.0.

Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Action classification in soccer videos with long short-term memory recurrent neural networks. In *International Conference on Artificial Neural Networks*, pages 154–159. Springer, 2010.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

François Chollet and Joseph J Allaire. Deep learning with r, 2018.

Carmen ME Colomer, David B Pyne, Mitch Mooney, Andrew McKune, and Benjamin G Serpell. Performance analysis in rugby union: a critical systematic review. *Sports Medicine-Open*, 6(1):4, 2020.

Molly Coughlan, Charles Mountifield, Stirling Sharpe, and Jocelyn K Mara. How they scored the tries: applying cluster analysis to identify playing patterns that lead to tries in super rugby. *International Journal of Performance Analysis in Sport*, 19(3):435–451, 2019.

Hayden Croft, Peter Lamb, and Simon Middlemas. The application of self-organising maps to performance analysis data in rugby union. *International Journal of Performance Analysis in Sport*, 15(3):1037–1046, 2015.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

Alex Graves. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*. Springer, 2012.

Cheng Guo and Felix Berkhahn. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*, 2016.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

Nicholas MP Jones, Stephen D Mellalieu, and Nic James. Team performance indicators as a function of winning and losing in rugby union. *International Journal of Performance Analysis in Sport*, 4(1):61–71, 2004.

Matthias Kempe, Andreas Grunz, and Daniel Memmert. Detecting tactical patterns in basketball: comparison of merge self-organising maps and dynamic controlled neural networks. *European journal of sport science*, 15(4):249–255, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2 (3):18–22, 2002. URL `https://CRAN.R-project.org/doc/Rnews/`.

Nazanin Mehrasa, Yatao Zhong, Frederick Tung, Luke Bornn, and Greg Mori. Deep learning of player trajectory representations for team activity analysis. In *11th MIT Sloan Sports Analytics Conference*, volume 2, page 3, 2018.

Jürgen Perl and Peter Dauscher. Dynamic pattern recognition in sport by means of artificial neural networks. In *Computational intelligence for movement sciences: Neural networks and other emerging techniques*, pages 299–319. IGI Global, 2006.

Mark Pfeiffer and Jürgen Perl. Analysis of tactical structures in team handball by means of artificial neural networks. *International Journal of Computer Science in Sport*, 5(1):4–14, 2006.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL https://www.R-project.org/.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Rajiv Shah and Rob Romijnders. Applying deep learning to basketball trajectories. *arXiv preprint arXiv:1608.03793*, 2016.

Brendan Teich, Roman Lutz, and Valentin Kassarnig. Nfl play prediction. *arXiv preprint arXiv:1601.00574*, 2016.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 2012.

Takamasa Tsunoda, Yasuhiro Komori, Masakazu Matsugu, and Tatsuya Harada. Football action recognition using hierarchical lstm. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 99–107, 2017.

Kuan-Chieh Wang and Richard Zemel. Classifying nba offensive plays using neural networks. In *Proceedings of MIT Sloan Sports Analytics Conference*, volume 4, 2016.

Neil Watson, Ian Durbach, Sharief Hendricks, and Theodor Stewart. On the validity of team performance indicators in rugby union. *International Journal of Performance Analysis in Sport*, 17(4):609–621, 2017.

Neil Watson, Sharief Hendricks, Theodor Stewart, and Ian Durbach. Integrating machine learning and decision support in tactical decision-making in rugby union. *Journal of the Operational Research Society*, pages 1–12, 2020.

Keane W Wheeler, David Mills, Keith Lyons, and Warrick Harrinton. Effective defensive strategies at the ruck contest in rugby union. *International Journal of Sports Science & Coaching*, 8(3):481–492, 2013.

Ronald J Williams and David Zipser. Gradient-based learning algorithms for recurrent. *Backpropagation: Theory, architectures, and applications*, 433, 1995.

Appendix

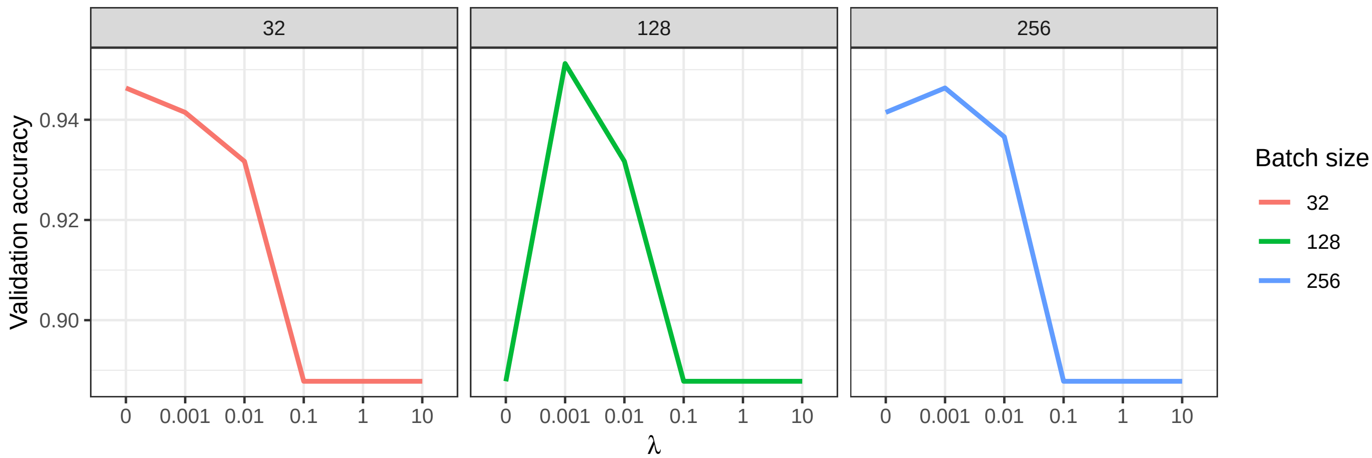

Figure .1: Validation Accuracy of LSTM model recognising the Exit Play (trained on imbalanced data)

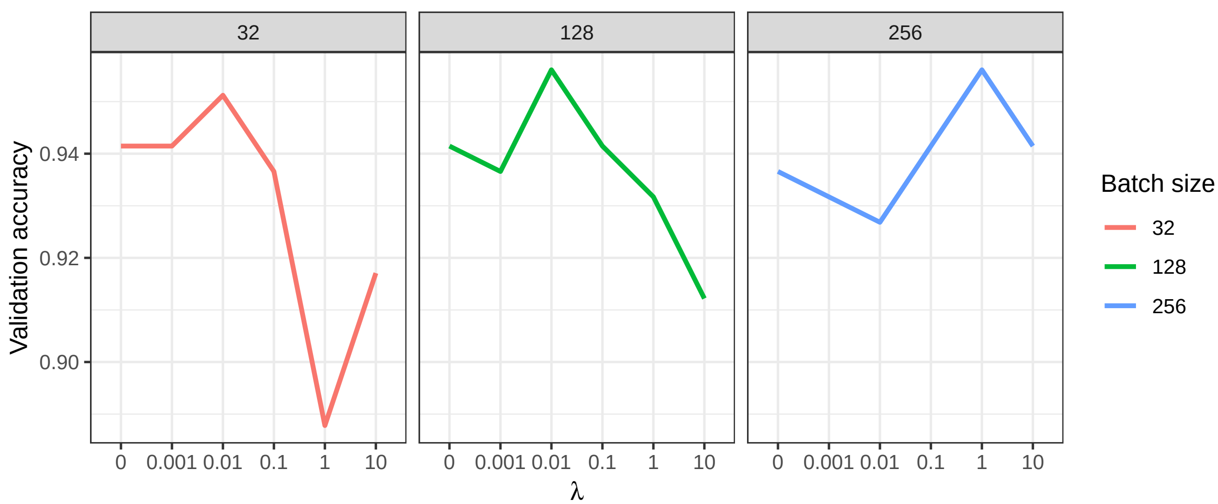

Figure .2: Validation Accuracy of Simple RNN model recognising the Exit Play (trained on rebalanced data)

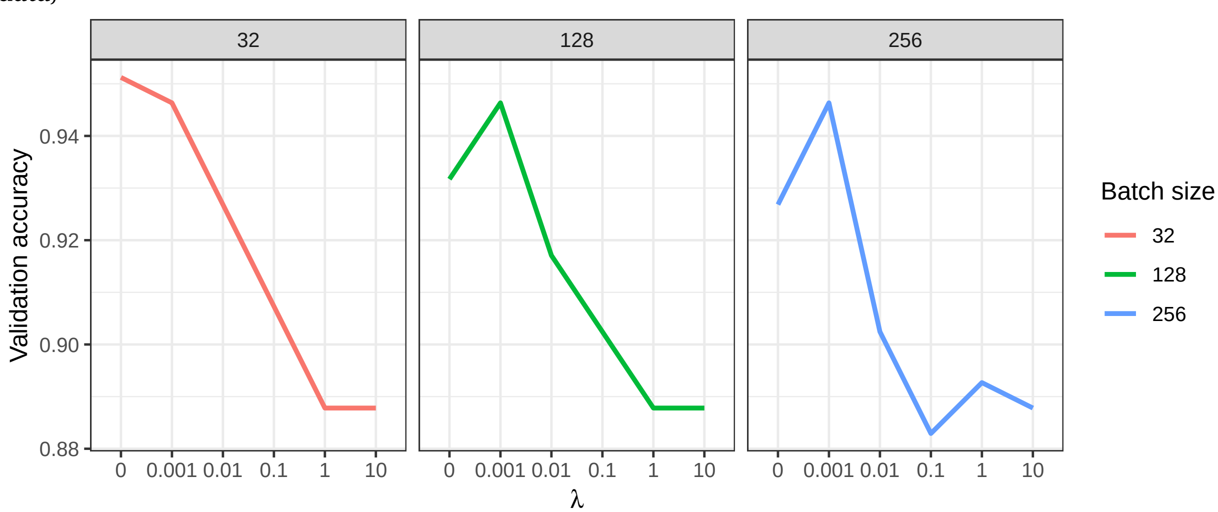

Figure .3: Validation Accuracy of LSTM model recognising the Exit Play (trained on rebalanced data)

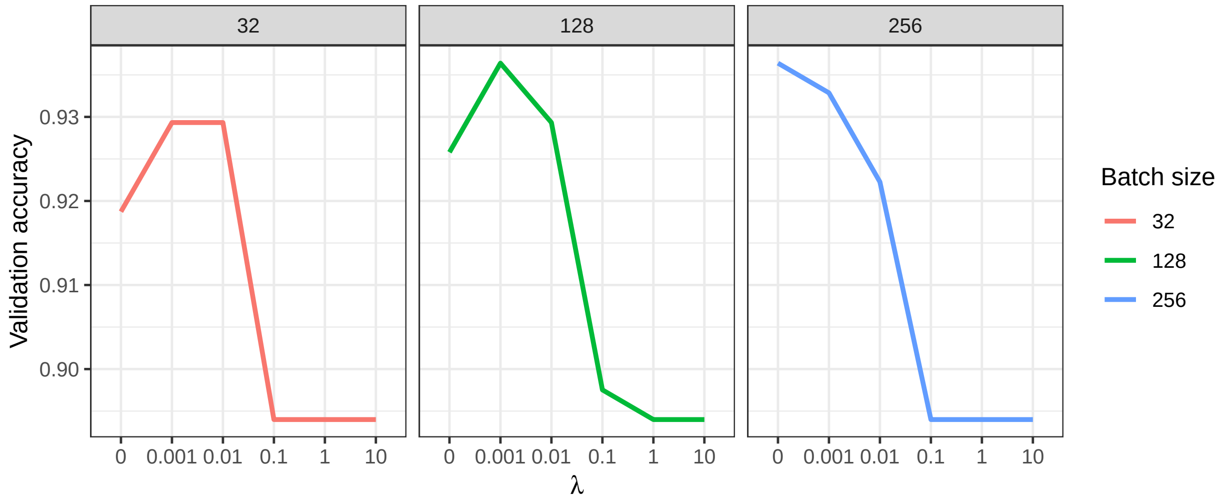

Figure .4: Validation Accuracy of Simple RNN model recognising the Forward Based Play (trained on imbalanced data)

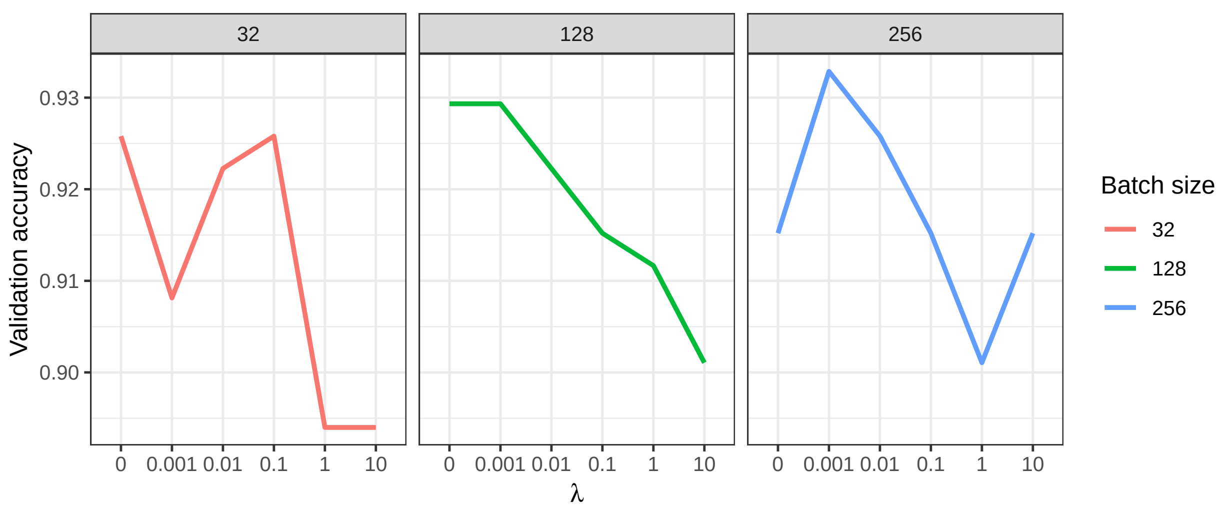

Figure .5: Validation Accuracy of Simple RNN model recognising the Forward Based Play (trained on rebalanced data)

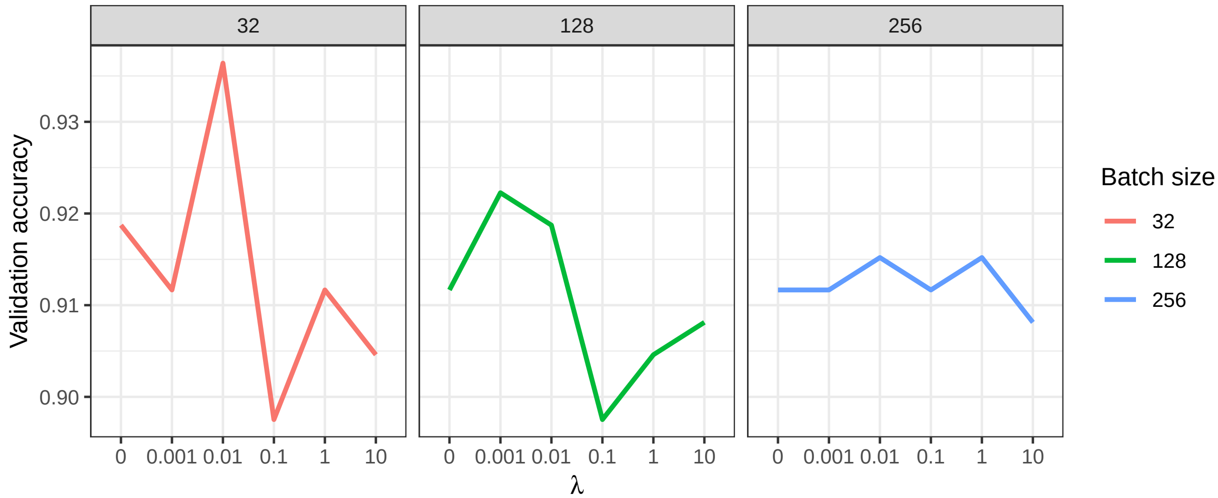

Figure .6: Validation Accuracy of LSTM model recognising the Forward Based Play (trained on rebalanced data)

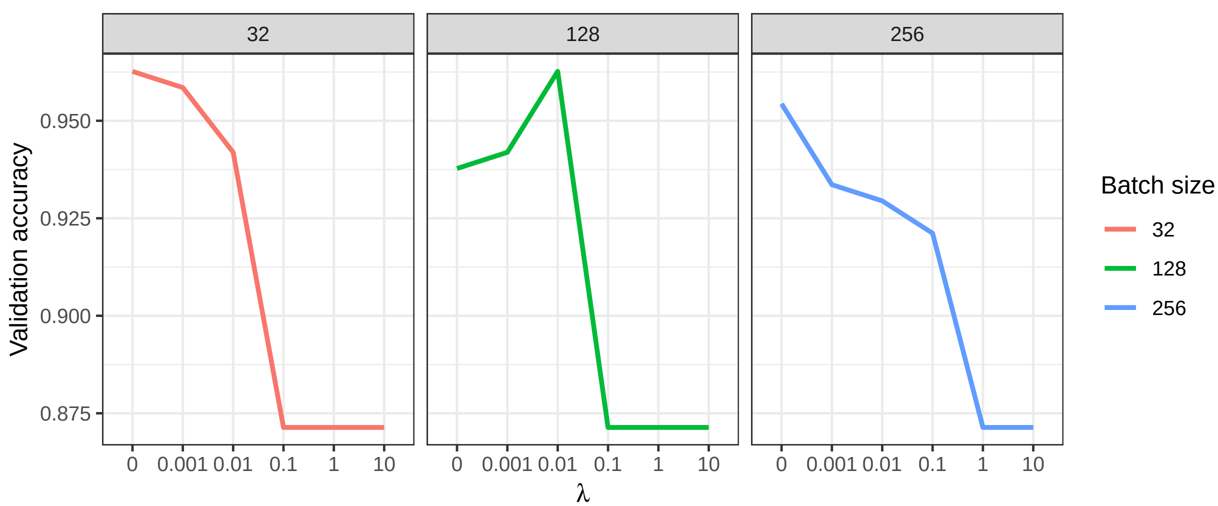

Figure .7: Validation Accuracy of Simple RNN model recognising the Wide-to-Wide Play (trained on imbalanced data)

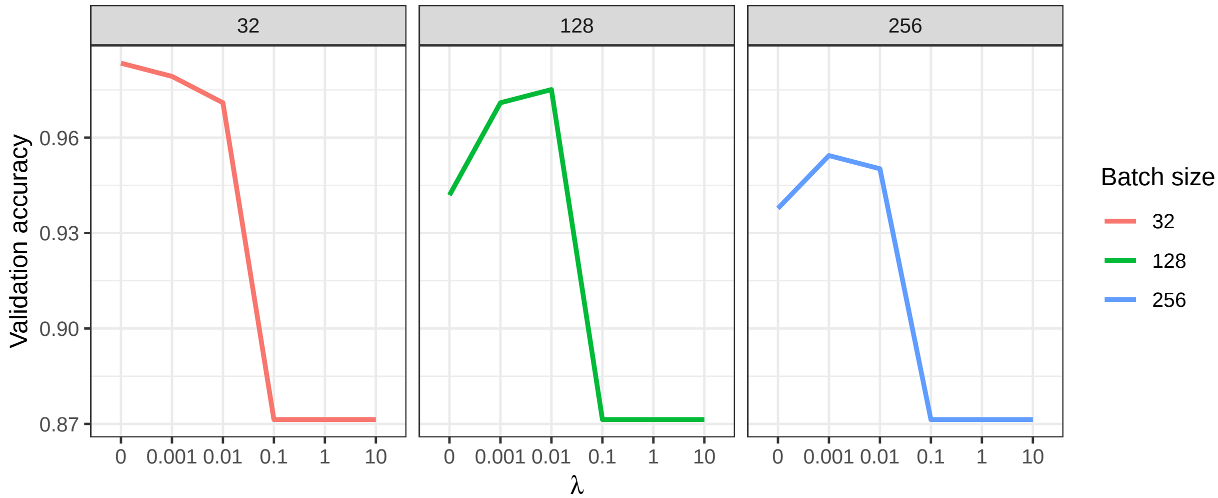

Figure .8: Validation Accuracy of LSTM model recognising the Wide-to-Wide Play (trained on imbalanced data)

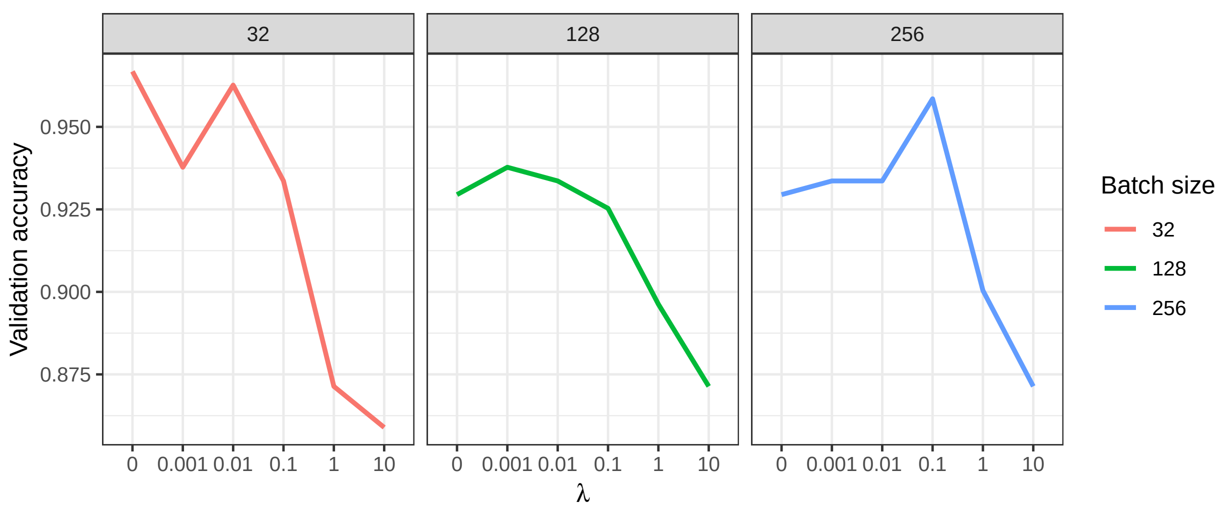

Figure .9: Validation Accuracy of Simple RNN model recognising the Wide-to-Wide Play (trained on rebalanced data)

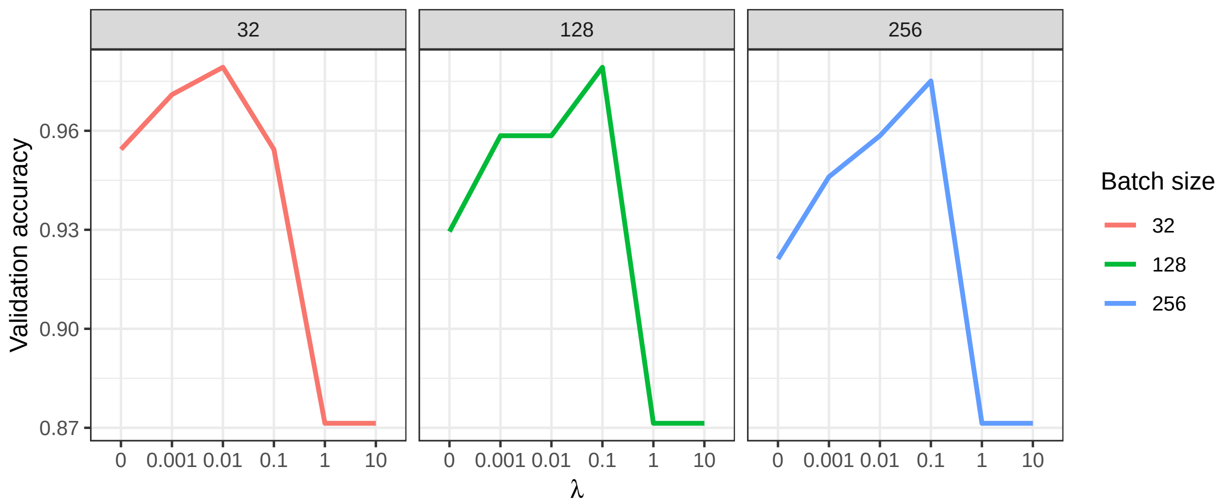

Figure .10: Validation Accuracy of LSTM model recognising the Wide-to-Wide Play (trained on rebalanced data)