Long short-term memory neural networks for predicting corporate credit ratings

Ali Chandoo

Dr Juwa Nyirenda

Master of Science

Data Science

Department of Statistical Sciences

University of Cape Town

2023

Acknowledgement

I begin by thanking Allah (s.w.t) for it is through his grace that everything happens.

I would like to extend my gratitude to my supervisor Dr Juwa Nyirenda who has been kind and extremely patient with me throughout this process. I would especially like to thank him for working through this project during odd hours and through many setbacks and not giving up on me.

I would also like to thank my mother, Shaimuna, for encouraging me to pursue an education and supporting me.

Abstract

Credit ratings are an important tool when assessing financial instruments and investments. The existing literature shows that long short-term memory (LSTM) neural networks are the best neural network to predict credit ratings, while random forests have been shown to perform better than regular neural networks. As at the beginning of this study, no study had compared the performance of LSTM and random forests despite their reported superior performance. This study compares the performance of random forests and LSTM neural networks in predicting corporate credit ratings in the USA using Standard and Poor's data. The study finds that while LSTM neural networks pose serious competition, random forests have a slight edge over LSTM neural networks, showing that it is still worth using older and simpler techniques in predicting credit ratings.

List of abbreviations and symbols

LSTM	Long short-term memory neural network
USA	United States of America
OLPM	Ordered linear probit model
RNN	Recurrent neural networks
S&P	Standard and Poor
OLS	Ordinary least squares
CART	Classification and regression trees
MOE	Mixture of Experts
NN	Neural network
RBF	Radial basis function
SVM	Support Vector Machine
MLP	Multilayer perceptron
ANOVA	Analysis of variance
MCAR	Missing completely at random
MNAR	Missing not at random
MAR	Missing at random
MICE	Multiple imputation by chained equations
SEC	Securities and Exchange Commission
OSPM	Ordered semiparametric probit model
BP	Back propagation
RF	Random forest
CO2	Carbon dioxide
BTT	Backpropagation through time
tanh	Hyperbolic tangent
PMM	Predictive mean matching
Val	Validation
Bagging	Bootstrap aggregation
$\int$	Hyperbolic tangent
σ	Sigmoid

Table of Contents

List of figures

List of tables

Chapter 1
Introduction

This thesis is concerned with the problem of predicting credit ratings using statistical and machine learning algorithms. This chapter gives a background to corporate credit ratings. The first section contains a brief introduction on credit ratings, a summary of its history to date and an explanation of credit ratings nomenclature. It then explains why credit ratings are important and how changes in credit ratings impact society. The next section outlines the main arguments and benefits of being able to predict credit ratings. The final section states the objective of this thesis.

1.1 Bond markets

The first ever openly available credit ratings were issued by John Moody in 1909 (White, 2010). Credit ratings are assessments of the relative likelihood that a borrower will default on their bond (Cantor and Packer, 1996). A bond is a structured promise to pay. It is a financial instrument used to raise debt. For example, a corporate credit bond is a financial instrument sold by a company to investors. In return, the company makes legal commitment to pay investors interest on the principal at fixed intervals, and to return the principal when the bond matures as described in the bond agreement (Choudhry, 2004).

Bond markets have existed for several centuries, from the 1600's in Western Europe and the 1800's in the United States of America (USA). There was essentially no need for a bond rating agency until the 1850's, as the majority of bonds were issued by countries or states. That is, they were sovereign, and there was little doubt in their repayment ability (Sylla, 2002). The USA however, had no sovereign debt by the year 1836. Instead, they had encouraged the development of corporations to develop their economy's productivity and advance their infrastructure (Sylla, 2002). This was a capital-intensive exercise which involved developing railroads and it was done at a large scale, often across different states. Securing capital for the corporations from banks in their respective states got difficult, and this prompted the creation of the largest corporate bond market in the world. Initially, all the bonds were for the

development of railroads, however with time bonds became accessible to all kinds of businesses (Sylla, 2002).

In 1909 John Moody began issuing public ratings for corporate railroad bonds to aid investors in picking the right bonds and making sense of the uncertainty. Moody's firm was successful on account of strong demand from investors. Seven years later, the Poor's Publishing Company and the Fitch Publishing Company were formed and they too began issuing credit ratings (White, 2010). This was followed by the formation of the Standard Statistics Company five years later which eventually merged with the Poor's Publishing Company to form the Standard and Poor's company. Today the three main credit rating agencies are Moody, Standard and Poor's and Fitch, and are commonly referred to as the big three (White, 2010).

Up until the 1970's, credit ratings were focused on the USA's corporate bond market. However, with globalization and financial development it eventually grew to rating corporate and sovereign bonds worldwide (Sylla, 2002). By 2013, the collective market share of the big three credit rating agencies was 95% with Moody and Standard and Poor's holding roughly equal shares and Fitch holding the lowest market share at 15% (Heinke and Steiner, 2001). The three rating agencies use letters and symbols to classify the quality of a bond (see Table 1 below for Standard and Poor's Global Long Term Issuer Rating's nomenclature).

Credit ratings are a standardised measure of credit risk. Broadly, bonds are either investment grade or speculative grade and ratings are issued over a wide range of financial instruments. The most commonly used are the short term and long term credit ratings. Credit ratings are not only issued for corporate bonds, they are also issued for sovereign bonds, mortgages and other similar financial instruments. Credit ratings alone are difficult to make sense of as they are just ordered letters and symbols. Thus, they are usually associated with a definition of creditworthiness. The definition explains the obligor's current and potential future financial situation and gives a recommendation on the obligor's ability to meet its obligations (see Table 1). The definition is still considered vague to base an investment decision on and thus, the credit rating agencies also publish data on the default and transition rates associated with each credit rating. This information may be used as a guide to assign a probability of default to bonds with similar ratings (see Table 2).

A credit rating alone does not determine the value of a bond as the price of a bond does not change directly due to a change in credit ratings; rather a change in credit ratings implies that the bond is either relatively more or less risky. A bond's price is determined by obtaining the present value of its expected cash flows using an appropriate discount rate - usually the yield-to-maturity (Seddik, 2015). The yield-to-maturity is the rate at which the bond's market price is equal to the present value of the bond's expected future cash flows (Seddik, 2015). Assuming the efficient market hypothesis is correct, then, when a bond's rating is lowered and its coupons are fixed, the increased riskiness of the bond will make it less desirable to investors. Due to demand and supply, the price of the bond will decrease and the yield-to-maturity will increase, and vice versa (McCarthy and Melicher, 1988).

Table 1: Standard and Poor's Global Long-Term Issuer Rating Scale.

Symbol	Description	Grade
AAA	An obligor rated 'AAA' has extremely strong capacity to meet its financial commitments. 'AAA' is the highest issuer credit rating assigned by S&P Global Ratings.	Investment Grade
AA	An obligor rated 'AA' has very strong capacity to meet its financial commitments. It differs from the highest-rated obligors only to a small degree.	
A	An obligor rated 'A' has strong capacity to meet its financial commitments but is somewhat more susceptible to the adverse effects of changes in circumstances and economic conditions than obligors in higher-rated categories.	
BBB	An obligor rated 'BBB' has adequate capacity to meet its financial commitments. However, adverse economic conditions or changing circumstances are more likely to weaken the obligor's capacity to meet its financial commitments.	
BB	An obligor rated 'BB' is less vulnerable in the near term than other lower-rated obligors. However, it faces major ongoing uncertainties and exposure to adverse business, financial, or economic conditions that could lead to the obligor's inadequate capacity to meet its financial commitments.	Speculative grade
B	An obligor rated 'B' is more vulnerable than the obligors rated 'BB', but the obligor currently has the capacity to meet its financial commitments. Adverse business, financial, or economic conditions will likely impair the obligor's capacity or willingness to meet its financial commitments.	
CCC	An obligor rated 'CCC' is currently vulnerable and is dependent upon favourable business, financial, and economic conditions to meet its financial commitments.	
CC	An obligor rated 'CC' is currently highly vulnerable. The 'CC' rating is used when a default has not yet occurred but S&P Global Ratings expects default to be a virtual certainty, regardless of the anticipated time to default.	

Source: (S&P Global Ratings, 2021a).

Table 2: Standard and Poor's observed default probabilities in % by rating for 2010-2020.

	AAA/AA/A	BBB	BB	B	CCC/C
2010	0	0	0.58	0.87	22.83
2011	0	0.07	0	1.68	16.42
2012	0	0	0.3	1.58	27.52
2013	0	0	0.1	1.65	24.67
2014	0	0	0	0.78	17.51
2015	0	0	0.16	2.42	26.67
2016	0	0.06	0.47	3.76	33.17
2017	0	0	0.08	1	26.56
2018	0	0	0	0.99	27.18
2019	0	0.11	0	1.49	29.76
2020	0	0	0.93	3.52	47.48

Source: (S&P Global Ratings, 2021b).

1.2 Why credit ratings are useful

Credit ratings have a significant impact on the value of a bond, the financial security of a business, and in turn the economy as a whole. A downgrade in credit ratings results in the loss of value for an investor, but because a downgrade warrants increased interest rates, the cost of capital for a borrower increases. The increase in the cost of capital makes it difficult for a business with a bad credit rating to acquire capital cheaply (Kim and Nabar, 2003). Corporations that are downgraded tend to have lower liquidity, and the fear of losing liquidity may force a firm to increase their leverage while borrowing is still cheap before the downgrade. This pre-emptive response may worsen their credit quality as the debt to equity ratio increases (Hung, Banerjee and Meng, 2016). Additionally, the increased cost of borrowing will negatively impact the corporation's performance and potential growth (Kim and Nabar, 2003). Hindered growth of a business can negatively impact the economy of a country by effectively reducing its capital investments and its overall productivity assuming that public investment remains constant (Nazmi and Ramirez, 1997).

Credit rating agencies also rate sovereign debt. Sovereign credit ratings form a small part of the overall credit rating industry, however, the effects of changes in sovereign credit ratings are realised over countries and regions (Ozturk, Namli and Erdal, 2016). A sovereign rating downgrade would imply that a country's debt is riskier and that the country may be less willing to repay their debts. Investors would aim to minimise their losses by divesting in that country, resulting in capital flight. To prevent this from happening, the monetary authorities may increase the interest rates on their bonds to make them worthwhile to investors, and this in turn increases the cost of capital and makes it difficult for a country to borrow money to finance their economic development (Yang and Zhang, 2011). Sovereign credit rating downgrades are also shown to have adverse effects on the currency's foreign exchange rate and its effect may linger or occur anytime within a ten day period before and after a downgrade (Brooks et al., 2004).

In anticipation of a sovereign credit rating downgrade, monetary and fiscal authorities may act quickly to mitigate its effects. This may be effective, however, there will still be damage through other channels. For example, some investors and institutions are legally bound to only invest in corporate bonds with or above a certain credit rating (Yang and Zhang, 2011). In the event of a downgrade they would be obliged to divest in the downgraded bond. Additionally, a 'herd-response' may be observed since the initial investors divesting may worsen the situation temporarily (Yang and Zhang, 2011).

Just as corporate credit rating downgrades can have an effect on the overall economy, there is some evidence to show that sovereign credit rating downgrades can have a negative effect on corporate credit ratings (Borensztein, Cowan and Valenzuela, 2013). Credit rating changes clearly have significant impacts on the economy and because of the spill over effects, the damage to the economy can be cyclical until an equilibrium point is reached (Giesecke and Weber, 2004).

The effects of credit rating changes do not only stop in the country that had a rating change, rather there may be spill over effects in neighbouring countries or in countries that have close economic and financial ties, such that the interest rates and asset values may be affected without that country having any rating event. This phenomena sometimes occurs in the Eurozone and emerging market countries (Ozturk, Namli and Erdal, 2016).

1.3 Why predict credit ratings?

The ability to predict credit ratings is important in a modern financial system. While in the past it may have been sufficient to do simple analyses of financial statements to understand the risk of a financial instrument or portfolio, this approach may not suffice anymore. This is because the complexity of financial markets have increased so much that it is impossible to manually assess creditworthiness of an instrument (Pettit, 2004). Credit ratings help fill this gap by providing a standardised measure of credit risk.

Although credit ratings are important in the capital market and almost all bonds are rated, only about twenty percent of issuers are rated on the New York Stock Exchange (Hwang, Chung and Chu, 2010). Bond ratings differ from issuer ratings in that the riskiness of a bond changes with many factors, including time. Thus, a bond is less risky the closer it is to maturity. However, one issuer may have several bonds and instruments and the overall creditworthiness of an issuer is not fully represented by any one of their bonds' ratings. Additionally, an issuer may have bonds that are rated but the issuer might not be rated. Having a model to predict and forecast issuer credit ratings would make it easier to invest in non-rated firms, thus creating new investment opportunities (Hwang, Chung and Chu, 2010).

The explanation for a lack of issuer credit ratings may lie in the history of credit ratings. In the past investors had to pay for ratings and the rating agency would provide them with their statistical analysis. The system has since changed and currently follows an issuer pay system (Sylla, 2002). Issuers pay the credit rating agency to review their financial and strategic position. The review involves not only providing the issuer with financial information but also expert opinion and outlook. This makes getting rated very expensive as it involves a lot of tasks and time and because of this, many firms are not able to get a rating. Even those that are rated may not be able to afford to revise their ratings (Huang, 2011).

In order to properly manage a portfolio, an investor or institution would need to timeously predict credit rating changes in their portfolios (Galil, 2003). Huang (2011) shows that the potential benefits of correctly forecasting credit ratings to a portfolio manager are: reduced volatility of returns, lower hedging costs as there is more awareness of risk and its patterns, and finally, all these effects may lead to overall higher profitability. Huang (2011) also

mentions that having an in-house model for predicting and forecasting credit rating can aid with better business management, as the firm could track its potential credit rating and use the information to steer itself in the correct direction. This is because the firm would be able to identify what variables and factors are driving its credit rating, and accordingly adjust its operations.

Furthermore, institutions that rely on the big three or are obliged to maintain a certain overall credit rating in their portfolios and often must make important decisions, such as selling off assets that may have been downgraded to junk status. Such actions are costly to them and to the issuer, and are generally done out of obligation, and sometimes without the full knowledge of what is driving the credit rating changes (Huang, 2011). Having their own credit rating prediction model could allow them the independence of taking on risk as they see fit.

This could be extended to other institutions such as banks and private investors and grant them more independence in their decision making. Global banking is regulated by several advisory boards, one of the most prominent organisations is the Basel Committee. The Basel Committee publishes accords that contain guidelines for banks and financial organisations to manage their overall risk (Goodhart, 2011). The most current is the Basel III standard, and it encourages for independent rating assessment and has discouraged overreliance on external credit ratings after the financial crisis of 2008/9. Firms and investors can benefit from having their own credit rating modelling and forecasting even if the issuer of the bond in question is already rated externally (King and Tarbert, 2011). This serves to add to the security and confidence of their investments.

1.4 Problem statement

Several methods for predicting credit ratings have been proposed since the first paper by Fisher (1959). The methods include both classical and modern. Prominent among the classical methods are: logistic regression, ordered linear probit model (OLPM) and discriminant analysis while the most popular modern methods include random forest and neural network based methods such as convolutional neural networks (CNNs), recurrent neural networks (RNNs) and long short-term memory (LSTM) networks.

A number of comparative studies have been undertaken to investigate the best performing model at predicting credit ratings. At the time this work began, comparative studies fell into two categories: those that compared random forest to at least one of the other algorithms excluding LSTMs (Carlenius et al., 2017; Wallis, Kumar and Gepp, 2019; Golbayani Wang and Florescu, 2020), and those that compared LSTMs to at least one of the other algorithms excluding random forests (Golbayani, Florescu and Chatterjee, 2020). Of the studies that included random forests, the studies found that random forest outperformed the other algorithms while the studies that included LSTMs found that LSTMs outperformed the competing algorithms. Ironically, so far, no study has compared the performance of random forest and LSTM given each method's reported superior performance in the category in which it was represented.

The objective of this thesis therefore is to compare the performance of random forest and LSTM at predicting credit ratings using the Standard and Poor's (S&P) long term issuer credit rating, an existing and widely accepted rating standard. The algorithms will be evaluated based on misclassification rate of test data. The study will also explore the importance of feature selection in credit rating prediction. Finally, the study will experiment with different numbers of previous time steps of a time series to use as input (that is: two most recent quarters, four quarters prior, etc.) in predicting credit ratings with a view of determining the number of prior time steps that lead to "optimal" predictions. The hypothesis is that all the information required to make acceptable predictions is contained in the most recent prior time steps of a time series.

The rest of the thesis is structured as follows, Chapter 2 discusses the existing literature on credit rating prediction, feature selection and the handling of missing data. This is followed by Chapter 3 which provides a comprehensive overview into the dataset. Chapter 4 then describes the models used in this study. Chapter 5 follows with an explanation of how the dataset is preprocessed and how the models are applied to it. Chapter 6 presents the results of this study which are then discussed in Chapter 7. The study is finally concluded by Chapter 8 which wraps up the findings of this study and proposes suggestions for future work.

Chapter 2
Literature review

2.1 Introduction

This chapter contains a review of existing literature relevant to corporate credit rating prediction, credit scoring models and similar financial applications. The chapter has six sections including this introduction. The first section contains a discussion on the pioneering papers in the field. The second section focuses on studies that perform comparative analyses on different credit rating prediction methods and gives an outline of the main methods used in this study. The review shows that there is indeed a gap and a lack of papers performing such comparative analysis, thus this section is supplemented with literature on credit risk prediction. The third section discusses the problem of missing data and how it is dealt with in the literature. It also discusses imputation of missing data using multiple imputation using chained equations, which is novel to studies on credit ratings. The fourth section contains a discussion of the feature selection processes and the appropriate features to be used. The final section contains a discussion regarding different testing criteria and the cost of misclassification.

2.2 Pioneering papers on credit rating prediction

There is vast literature on predicting credit risk which can be categorised in terms of how credit risk is quantified, such as but not limited to rankings and default probability. The earliest papers focus on predicting credit quality and bankruptcy rather than credit ratings. As credit ratings got more relevant and popular, the literature grew with it, especially literature focused on identifying the drivers of credit ratings and how to predict them.

The earliest study aimed at identifying the drivers of credit rating was carried out by Fisher (1959). Fisher (1959) used multiple regression to investigate the impact of features associated with default and market risk on variation in domestic industrial bond yields. Some of these features are equity to debt ratio, period of solvency, volume of trading and bonds outstanding. Fisher (1959) found that all features considered were statistically significant predictors. This finding was significant as there was scepticism among the public to believe that historic data

could inform credit ratings. Horrigan (1966) too found that by using accounting information and financial ratios that specifically focused on liquidity and profitability, credit ratings could be predicted using multiple regression with about 60% accuracy for Moody's and Standard and Poor's bond ratings. Horrigan (1966) used ordinary least squares to estimate the parameters of the model. The same approach was used by West (1970).

However, McKelvey and Zavoina (1975) and later, Kaplan and Urwitz (1979) pointed out that OLS assumptions are violated when the dependent variable is ordinal, leading to heteroscedasticity and that the sum of the errors is neither zero nor displays a normal distribution. As an alternative, Pinches and Mingo (1975) and Altman and Katz (1976) proposed the use of discriminant analysis to predict credit ratings despite its own weaknesses. For instance, discriminant analysis requires the dependent variable to be categorical in nature. However, credit ratings are ordinal. Additionally, discriminant analysis requires the fulfilment of difficult assumptions such as multivariate normality and its estimates are weakened by the presence of multicollinearity in the explanatory variables (Altman and Katz, 1976; Kaplan and Urwitz, 1979).

Kaplan and Urwitz (1979) performed the first comparative study of statistical models for predicting corporate credit ratings. They used an ordered linear probit model (OLPM) as it accounts for the ordinal nature of the credit rating variable (Matthies, 2013). They compared performance of OLPM to that of OLS based models using features from the study of Horrigan (1966). They found that OLS based models performed slightly better with a test accuracy of 55% while OLPM yielded an accuracy of 50% only.

Although the methods mentioned above may not be very relevant to this study, they form the basis for performing comparative analyses in this field. Four main points can be learnt from these studies that are crucial to performing a credible comparative study. Firstly, the necessity of testing the models on the same test data as accuracy rates will vary greatly over different datasets. Secondly, using the correct model, such that it can accommodate the complex nature of credit ratings. Thirdly, applying an adequate method for feature selection. When features are selected using either financial theory or an appropriate feature selection method fewer features are needed to acquire a similar or higher test accuracy. Finally, selecting an appropriate test criteria and measure of accuracy such that real world inference can be made. This is sometimes referred to as the cost of 'misclassification'.

Following these principles, a comprehensive study was carried out by Ederington (1985), where he compared the performance of OLS and OLPM, discriminant analysis, and an unordered linear logit model at predicting corporate credit ratings. Ederington (1985) found that the OLPM was the best performing model. Similarly, Matthies (2013), observed that ordered response models are the best performing "classical methods" and that they are widely used in financial applications. They are also often used as a baseline model when comparing the performance of modern machine learning algorithms in credit rating prediction and financial risk modelling (Ye, Liu and Li, 2008; Novotná, 2012; Ozturk, Namli and Erdal, 2016). Moreover, in some cases, ordered response models have been modified to use non-linear and semi-parametric functions to predict credit ratings (Hwang, Chung and Chu, 2010).

2.3 Tree based models vs neural networks

Tree based models are uncommon in mainstream credit rating predictions, however they are used extensively in credit scoring models, default prediction, bankruptcy prediction and other financial modelling applications. The performance of tree-based models however, seems to be a mixed bag in financial applications. While some studies claim tree-based models to be the best performing models, other studies have reported that even simple linear models outperform tree based models. A review of some relevant literature on this issue is presented in the next paragraphs.

Hajek and Michalak (2013) reviewed 18 studies on corporate credit rating prediction. Hajek and Olej (2014) found that trees with pruning outperformed multilayer perceptron (MLP) neural networks, radial basis function (RBF) neural networks and support vector machines (SVM) when predicting between investment and non-investment grade S&P credit ratings. Meanwhile, Satchidananda and Simha (2006) compared the performance of logit models and classification and regression trees (CART) in predicting loan defaults for Indian banks, and found that CART outperformed logit models and did so with fewer independent variables.

West (2000) compared the accuracy of classical methods such as linear discriminant analysis and logistic regression to modern statistical learning methods including but not limited to neural networks and CART in predicting credit scores. West (2000) found that the mixture of expert (MOE) neural networks was the best performing model while the logit model was

slightly more accurate on average. Non-parametric models such as K nearest neighbours and CART were found to be the worst performing models with CART taking the lower rank when tested on both German and Australian credit data. Novotná (2012) also found that logit models and discriminant analysis outperformed CART in predicting credit ratings of European companies.

Wang et al. (2011) has attributed two reasons for the poor performance of CART models in financial applications. The first reason is that CART models are sensitive to noise and the second is that CART models are sensitive to redundant explanatory variables.

Some of the flaws of decision tree algorithms can be remedied by using ensemble strategies. Ensemble strategies involve using multiple learning algorithms on the training data and combining them to solve the problem presented (Zhou, 2009). The most common tree-based ensemble methods include random forests, random subspaces, boosting and bootstrap aggregation (bagging). These methods have been compared on multiple datasets and their performance tends to vary across different datasets. However, they all result in an improvement in accuracy relative to regular decision trees (Banfield et al., 2006). Ensemble techniques in general work to train a decision tree on different subsets of the data or using different subsets of the explanatory variables or by weighting the explanatory variables iteratively. Randomisation helps correct for the trees' sensitivity to noise and outliers essentially by creating new data and multiple classifiers (Dietterich, 1999). These ensemble techniques have been applied to predicting credit ratings and creating credit scoring and general financial risk models.

Wang et al. (2012) compared the accuracy of several machine learning algorithms in predicting German and Australian credit risk. While CART was the worst performing model, random forests outperformed MLP, discriminant analysis, logit models and radial basis function (RBF) networks. Random forests have also been used in a semi-supervised learning algorithm to predict Moody's and Standard and Poor's corporate credit ratings in Japan (Saitoh, 2016). Additionally, Ozturk, Namli and Erdal (2016) found random subspace and random forest models to be the best performing models for predicting sovereign credit ratings relative to MLP. Carlenius et al. (2017) showed that random forest outperformed MLP in predicting corporate credit ratings for locally listed Norwegian firms. Addo, Gueggan and Hassani (2018) compared the performance of logit, random forest, gradient boosting and several deep learning

MLP models and found that random forests and gradient boosting models outperformed all other models significantly in predicting default for bank loans. Random forest and gradient boosting models maintained their lead in accuracy even when the explanatory variables were limited to the best ten, while the performance of the deep learning models dropped significantly and became unstable (Addo, Guegan and Hassani, 2018).

Wallis, Kumar and Gepp (2019) compared the accuracy of several machine learning techniques including but not limited to: MLP, boosted trees, support vector machines (SVM) random forest, logistic regression, linear discriminant analysis in predicting and forecasting Moody's Long Term Issuer ratings for American (USA) corporations. They found that random forests outperformed all the other models, while SVM and MLP came in second place with similar accuracy. Meanwhile, Golbayani, Florescu and Chatterjee (2020) showed that random forests outperform SVM and MLP when predicting S&P credit rating.

The major downside of neural networks is the risk of overfitting the data. Overfitting is generally avoided by using the right type of architecture (Livingstone, Manallack and Tetko, 1997). However, choosing the right type of architecture is difficult and can be computationally costly. Thus, most researchers use multilayer perceptrons (MLP) in their models due to their simplicity (Arouri et al., 2014). However, MLPs have been shown to have lower accuracies in financial applications (Wilamowski, 2009).

As a result, more sophisticated neural network architectures such as Convolution Neural Networks (CNNs), recurrent neural networks (RNNs) and long short-term memory (LSTM) neural networks have been used in financial applications and have been shown to outperform multilayer perceptrons (Zhou and Dai, 2015; Fu, et al., 2016; Chen, Althelaya, El-Alfaly and Mohammed, 2018; Sun, Wei and Wang, 2018). Golbayani, Wang and Florescu (2020) were the first to compare the performance of LSTM, CNN and MLP neural network architectures at predicting the Standard and Poor's corporate credit ratings for firms in the financial, health and energy sectors of the USA. They found that LSTM models outperformed MLPs and CNNs.

2.4 Feature selection

Once the models to be used in a comparative study have been identified, the next task is to identify the features to be used in the data and to choose the feature selection process. Initially the features were selected arbitrarily (Horrigan, 1966), and this was improved upon in the subsequent two studies (West, 1970; Pinches and Mingo, 1973).

In more recent literature Hajek and Michalak (2013) reviewed sixteen studies that built models to predict S&P and Moody's corporate credit ratings for firms in the USA. Of these studies, eight of them used a feature selection method. The most popular feature selection method was analysis of variance (ANOVA). It is not known if this is the best or most accurate method as the test accuracies are not directly comparable since the studies differed in terms of the data, the models used, and the time period.

In a review of one hundred and thirty papers focusing on machine learning techniques in financial prediction applications, Lin, Hu and Tsai (2011) state that the best feature selection method is unknown as a large number of them have been applied in different applications and on different datasets.

However, it can be argued that feature selection is important and should be used, nonetheless. Given the immense amount of data available and the large number of financial ratios and variables that are ever so slightly different, it can be very computationally expensive and inefficient to run models with complete feature sets. Feature selection can help reduce computation time, memory, and storage requirements (Chen, 2012).

The benefits of feature selection are not only computational as feature selection can also make it easy to collect and interpret data (Chen, 2012). Dimensionality reduction is another way of obtaining fewer features that make it easier to create visualizations that could aid experts and clients to understand how the rating process works, thereby increasing the practical interpretability of the model and the data (Chen, 2012).

Golbayani, Wang and Florescu (2020) compared the performance of different neural networks (CNN and MLP) in predicting corporate credit ratings by examining two cases. The first

involved preselecting the best features while the second case involved using all the features and letting the algorithms select the best features by weighting and by variable importance. They found that allowing the algorithm to select features through its own mechanism showed better results than preselecting features. It should be noted that they imputed all the missing data points with a zero and suggested that providing more processing data might allow for better predictions. Ironically, although the LSTM model was their best performing model (see above) they did not include it in this experiment.

Modern neural network algorithms such as the LSTM neural network can adequately determine the features that are most important and as such LSTM neural networks have been used for feature selection and have been found to be useful in high dimensional bioinformatics datasets (Agbehadji et al., 2018). Similarly, random forests too have been shown to be good at feature selection. For example, Yeh, Lin and Hsu (2012) used random forests to select the best features for multiple models used to predict Moody's corporate credit ratings, this selection of features by random forest led to improvements in prediction accuracy.

The process of assigning a credit rating is proprietary knowledge and is unknown to the public, thus, it is rather difficult to determine what features are used in the prediction from the get go (Hajek and Michalak, 2013). Hajek and Michalak (2013) explain that two main categories of financial data are used in predicting corporate credit ratings, that is, business position and financial indicators. The business position category aims to define the size of the company, its reputation, market capitalization, management expertise and industry risk (Hajek and Michalak, 2013). The financial indicators are explained in more detail and aim to give a snapshot of a company's financial position. The financial indicators measure profitability, liquidity and leverage (Hajek and Michalak, 2013). Hajek and Michalak (2013) conducted a comprehensive study of the features used to predict corporate credit ratings. They used over 80 features for US data and about 40 for European data; these features are listed in Table 1 of the appendix. Some specific features used in earlier studies are as follows: current ratio (Delahunty, 2004; Kim, 2005), total assets, total debt/total assets, net margin, interest coverage (Brennan and Brabazon, 2004), total liabilities, cash flow, net income (Huang et al., 2004) and dividend yield (Hajek, 2012).

Golbayani, Wang and Florescu (2020) for example, use all the data available to them from Bloomberg and Compustat, with a total of 332 features covering 30 companies. That is, over

four times the number of features as in the comprehensive list offered by Hajek and Michalak (2013). The use of all features in the dataset to make predictions were also used in some earlier studies (Garavaglia, 1991; Huang et al., 2004).

This study will combine the approaches mentioned in the previous paragraph to guide the data collection process, clean the dataset, and select the appropriate features. The process of creating a complete dataset is discussed in detail in the data overview section.

2.5 Missing data and imputation

There are two main approaches used to deal with missing data in credit rating datasets. The first is to delete the data or to only select data without missing observations (Hwang, Chung, and Chu, 2010) and the second is to impute the data. In the current literature imputation has been approached in two ways, zero imputation (Golbayani, Wang and Florescu, 2020), and median value imputation (Hajek and Michalak, 2013).

Dropping incomplete observations can lead to several problems. For example, the sample size is reduced. Additionally, it can lead to loss of statistical power and bias if crucial variables are removed (Peng et al., 2006). Similarly, mean, median and zero imputation can also create problems in the dataset. Mean imputation can distort the distribution of a variable and alter its relationship with another variable as their correlation is reduced (Gelman and Hill, 2006). Meanwhile, zero imputation is shown to create sparsity in the data and reduces the performance of learning algorithms such as neural networks (Yi et al., 2019).

Before trying to correct for missingness, it is important to understand the missingness in the dataset. There are two aspects to consider when analysing missing data. The first is the rate of missingness, and the second is the type of missingness. There are several estimates on the rate of missingness that are acceptable. For example, five percent (Schafer, 1999), and ten percent (Bennet, 2001). However some studies define different conditions. For example, McNeish (2017) shows that missingness rates of up to thirty percent are acceptable depending on the sample size and imputation method used. Finally, Tabachnick, Fidell and Ullman (2007), state that the rate of missingness is less important than the type of missingness in identifying whether the missing data is a significant problem.

There are three types of missingness as explained by Rubin (1976). Missing completely at random (MCAR), missing at random (MAR) and missing not at random (MNAR) and they have different implications. MCAR means that the data missing has no underlying connection to the data that is observed in the dataset, and thus, in terms of the dataset, the missing data is just lost information and can be ignored. MNAR is missing data which cannot be explained in terms of the data, and as such there is no real solution but to get more data that might attempt to explain it. Finally, MAR is data missing that can be explained by the data present in the dataset.

In the presence of missing data that is MAR, the missing points can be safely imputed. There are several methods of imputation however multiple imputation by chained equations (MICE) has been shown to be one of the best imputation methods across different studies (Ambler, Omar and Royston, 2007; Baneshi and Talei, 2012). Multiple imputation by chained equations (MICE) is a process of imputing missing values by identifying relationships with the existing data. MICE does not heavily distort the distribution of the variables and is shown to be accurate at predicting missing values. It is flexible in terms of the algorithm that is used and thus can tackle different types of data including categorical and continuous. However, it is computationally expensive and operates on the assumption that the data is MAR (Buuren and Groothuis-Oudshoorn, 2010; Royston and White, 2011).

MICE can use several algorithms to impute the missing values. Heidt (2019) shows that predictive mean matching (PMM) and classification and regression trees (CART) were the best performing algorithms even outperforming random forests. However, Chhabra, Vashisht and Ranjan (2017) show that the results from several algorithms including PMM, CART and random forests perform similarly. This study will use the CART algorithm.

MICE is an iterative process and the optimal number of iterations is unknown. However there are some rules of thumb suggested in the literature. The first is to use the missingness percentage as the minimum number of iterations to run (White, Royston and Wood, 2011). A second suggestion is to use five (Royston and White, 2011) or to use ten (Raghunathan, Solenberger and Van Hoewyk, 2002). A third suggestion is to use a much larger number such as forty as this may increase performance (Graham, Olchowski and Gilreath, 2007). However this is extremely computationally intensive and may be both impractical and unnecessary (Azur et al., 2011)

It can be difficult to accurately understand missingness in a big dataset as a very large number of patterns can arise when assessing the relationship between variables. Additionally, in real life scenarios the missing data is rarely accessible. This means that it is arguably impossible to truly confirm if data is MAR (Gelman and Hill, 2006). However, it is crucial to explore the data as best as can be to avoid creating problems when imputing and applying statistical learning models to it. Fortunately, there is some guidance in the literature to assist in making the best decision with the information available.

McKnight et al. (2007) explain simple methods of identifying patterns in missing data to classify the type of missingness. The first step in handling missing data is to list the missing data according to case and variable to observe obvious missingness levels. This should be followed by a preliminary cleaning. The data can then be analysed to find patterns in missingness. McKnight et al. (2007) suggest a method of using indicator values for each missing point and then tabulating these values to identify what if any type of pattern exists in the missing values. This study will follow the suggestions made by McKnight et al. (2007) in determining the type of missingness in the data.

2.6 Cost of misclassification

The cost of misclassification refers to how the target variable is defined. The target variable can be all the individual levels of credit ratings or can be grouped into different categories such as ‘investment grade’ or ‘junk’. In other words, is the prediction being made for all levels of credit ratings or is the prediction meant to differentiate between groups of ratings. The definition of the target variable will have an impact on the accuracy and misclassification rate of the models.

Many studies tend to focus on predicting between highly ranked bonds and lower ranked bonds (Hwang, Chung and Chu, 2010; Shin and Han, 2001; Huang et al., 2004). A smaller number of studies (Golbayani, Florescu and Chatterjee, 2020; Golbayani, Wang and Florescu, 2020) predict all the distinct classes available in the dataset, however, they predict different sectors individually and this approach is not generalisable to newer observations that might be classified a rating that did not exist in their historic dataset. This study will follow the approach followed by the majority of predicting between highly and lowly ranked bonds.

Additionally, this study will attempt to predict two quarters ahead at the minimum. This is because the Securities and Exchange commission (SEC) requires companies to report quarterly financial data via form 10-Q, 40-45 days after the end of their quarter. Therefore a prediction window of 2 quarters is sufficient as there will be updated financial information before the end of the second quarter and a more accurate prediction can be made with newer data. (U.S. Securities and exchange commission, 2022).

Table 3 summarizes the most relevant studies to this one, showing empirical performance based on different models, the number of classes used in their prediction and whether or not they employed feature selection methods, this data will set the basis of comparison when discussing the results obtained from this study.

Table 3: Table showing empirical results from historic papers predicting on S&P ratings in the USA relevant to this study.

Authors	Method	Feature Selection	Classes	Misclassification (%)
Hwang, Chung and Chu, 2010	OLPM	Yes	3	24.00
Hwang, Chung and Chu, 2010	OSPM	Yes	3	18.90
Huang et al., 2004	SVM	Yes	5	19.62
Huang et al., 2004	BP-NN	Yes	5	19.25
Garavaglia, 1991	BP-NN	Yes	3	15.10
Hajek and Olej, 2014	CART	Yes	2	17.59
Hajek and Olej, 2014	MLP	Yes	2	24.19
Golbayani, Wang and Florescu, 2020 (Healthcare sector)	LSTM	No	Not fixed[1]	18.65
Golbayani, Wang and Florescu, 2020 (Energy sector)	LSTM	No	Not fixed[1]	11.20
Golbayani, Wang and Florescu, 2020 (Healthcare sector)	CNN	No	Not fixed[1]	27.70
Golbayani, Wang and Florescu, 2020 (Energy sector)	CNN	No	Not fixed[1]	21.32
Golbayani, Wang and Florescu, 2020 (Healthcare sector)	MLP	No	Not fixed[1]	30.50
Golbayani, Wang and Florescu, 2020 (Energy sector)	MLP	No	Not fixed[1]	22.96
Golbayani, Florescu and Chatterjee, 2020 (Healthcare sector)	MLP	Yes	Not fixed[1]	23.37
Golbayani, Florescu and Chatterjee, 2020 (Energy sector)	MLP	Yes	Not fixed[1]	21.81
Golbayani, Florescu and Chatterjee, 2020 (Healthcare sector)	RF	Yes	Not fixed[1]	17.03
Golbayani, Florescu and Chatterjee, 2020 (Energy sector)	RF	Yes	Not fixed[1]	15.55

1. Number of classes were the number of distinct classes present in the study's dataset, varies from 5 to 19
2. BP – Back propagation, MLP – Multilayer perceptron, RF – Random forest, CNN – Convolutional neural network, LSTM- Long short-term memory, CART – Classification and regression trees, OLPM – Ordered linear probit model, OSPM – Ordered semiparametric probit model

Chapter 3
Data overview and exploratory data analysis

3.1 Introduction and overview of the dataset

This chapter describes the data used in this study. It outlines the data collection process, cleaning process, the way missing data was handled and the method of imputation that was used to create a complete dataset suitable for analysis.

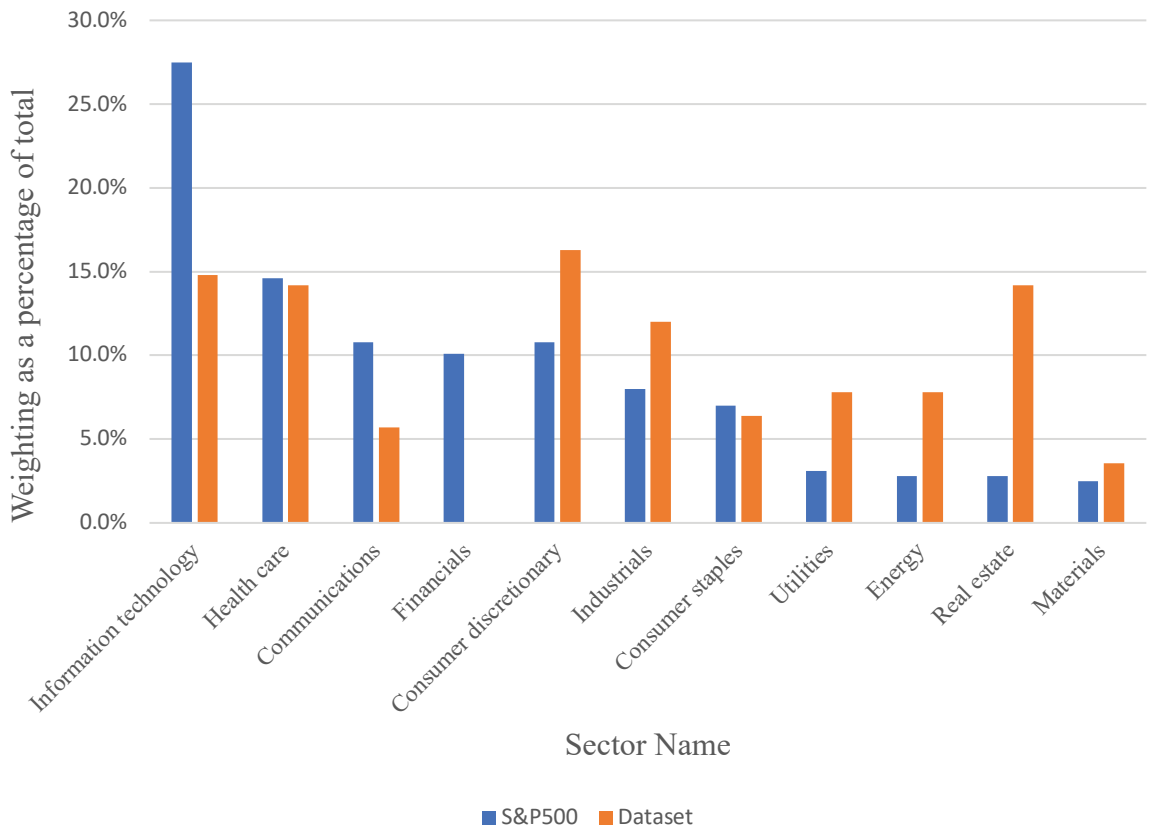
3.2 Data collection process and description

The data used in this study was collected from Bloomberg. Quarterly data was preferred to yearly data because of the following reasons. First, accounting data is publicly available quarterly on Bloomberg so that the latest data is used to make predictions. Second, long term ratings do not change frequently, nor do they change at fixed intervals. For example, once every year, long term ratings can change at any time. This means that if there is a credit rating change in the middle of a year, it would only be reflected in the following year (if yearly data is used), and there might be slight mismatches between what the accounting data is indicating and the credit rating; whereas, if quarterly data is used, at most the information lag can be three months.

The next step was to select firms that would be included in the raw dataset. In an attempt to make the dataset representative of the overall stock market in the USA, the S&P 500 index was used as a rough guide in the selection of firms. The S&P 500 index comprises stocks from the top 500 largest firms in the USA. The index is created by weighting different stocks from different sectors. Figure 1 compares the weightings of each sector in the S&P 500 index with the weightings of each sector in the data collected. Three criteria were used in the selection process. Firstly, firms were selected proportionately from each sector to ensure that their weightings were similar to those of the S&P 500 index. Secondly, only firms that had credit ratings from the first quarter of 2010 to the fourth quarter of 2019, and experienced at least one credit rating change in the same period were considered for selection. A total of 166 firms that met the criteria were selected each with over 300 variables which included the S&P long term

issuer rating and non-financial variables such as Environmental disclosure score and Total CO2 emissions.

Figure 1: Bar graph showing sector weightings for S&P 500 index compared to the data collected in this study.



Source: (S&P Dow Jones Indices, 2020).

The dataset cannot be in the exact proportions as the S&P 500 index for two main reasons. The first reason is that the financials sector is excluded from this study, as it follows different rules, regulations, and accounting conventions. As a result, many explanatory variables that are present in the data for other firms are missing in the data for firms in the financial sector (Hwang, Chung and Chu, 2010). The second reason the dataset cannot match the exact proportions of the S&P 500 index is that when creating the dataset, only firms that experienced at least one change in their credit rating over the period of the dataset were selected. For example, many large IT firms were excluded as they had the highest rating and experienced no change for the duration of the dataset. This was done to create a more representative sample, thereby allowing for the models developed to be relevant across all rating levels and all sectors. Despite the mismatches, the dataset could be said to be useful and representative.

The S&P long term issuer credit rating was selected in the dataset because (1) accounting data and financial ratios are generally updated quarterly (publicly available), and thus makes sense to use a rating that changes at a similar or slower pace such that the information can be captured. (2) the long term rating is only assigned to about 20% of listed firms. This means that being able to predict long term ratings is beneficial for most firms who are not rated (Hwang, Chung and Chu, 2010). (3) long term ratings are available for a longer period. This allows the dataset to have data over a longer period.

The dataset contains credit ratings ranging from AAA to CCC, and because most ratings are in between the rating scale; the lower ratings are least frequent. As explained in the introduction and literature review, there are twenty-one different credit rating intervals. Of these twenty-one rating levels, seventeen are present in the dataset.

The dataset contained a large amount of missing data. For example, values for non-financial variables that could have provided an interesting aspect to the analysis such as Environmental disclosure score and Total CO2 emissions were missing throughout the entire dataset. The dataset had 241843 missing points, which is roughly 21% of the entire dataset. The aim of this process is to end up with a dataset that contains as many firms as can be retained without having too many missing observations and to contain all the important variables to match closely most of the previous studies.

3.3 Data cleaning process

The first step in the data cleaning process was to deal with missing data. To identify the how and where the data was missing, a function in R was written to determine the missing rate by case and variable. In this case, this meant missing by firm and feature. Additionally, it considered missingness by sector and overall missingness in the dataset (McKnight et al., 2007). In the original dataset, the largest missing rate was observed when data is grouped by variables - up to a hundred percent missing rate was obtained for some variables, whereas when grouped by firms, the highest missing rate was less than fifty percent.

To correct for the large missing rate, any variable that was missing in more than half the observations was removed. In this step, thirty variables were removed, reducing the number of

total missing points to 76911, representing 8% of the entire dataset. The overall missingness levels for most of the firms also reduced, with most firms recording a missingness level of under ten percent. These changes are shown in Figures 2 and 3. Figure 2 shows the removal of features that were missing in at least fifty percent of cases. Figure 3 shows the impact of removing these features on the missing rate by firm.

Figure 2: Bar graph showing the level of missing data by feature before and after removing the most missing features.

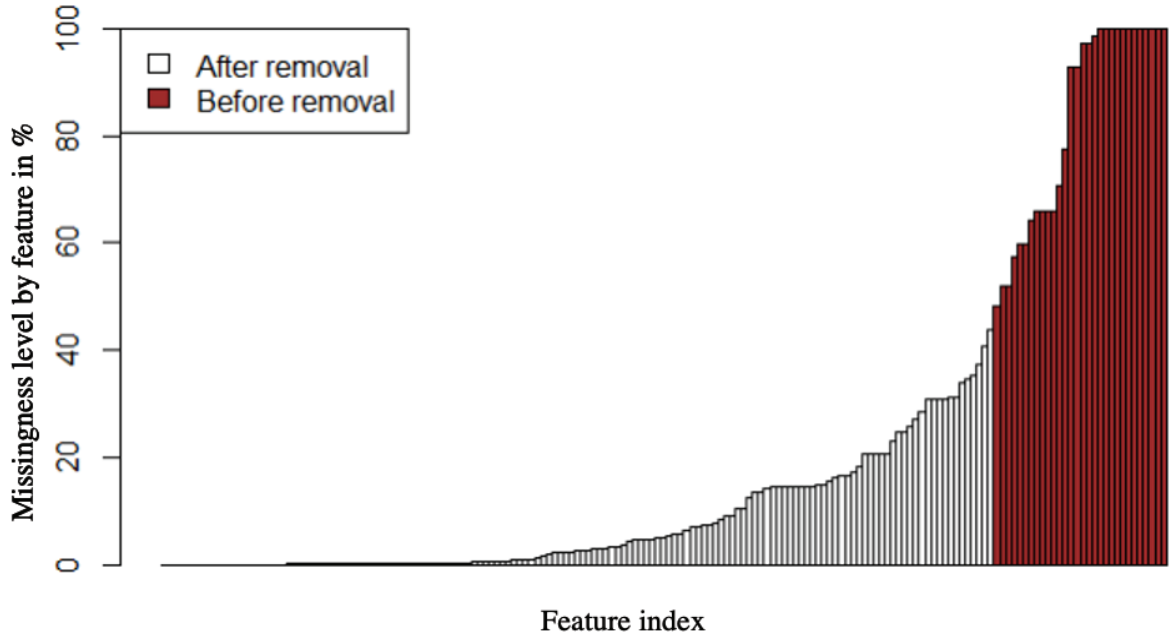
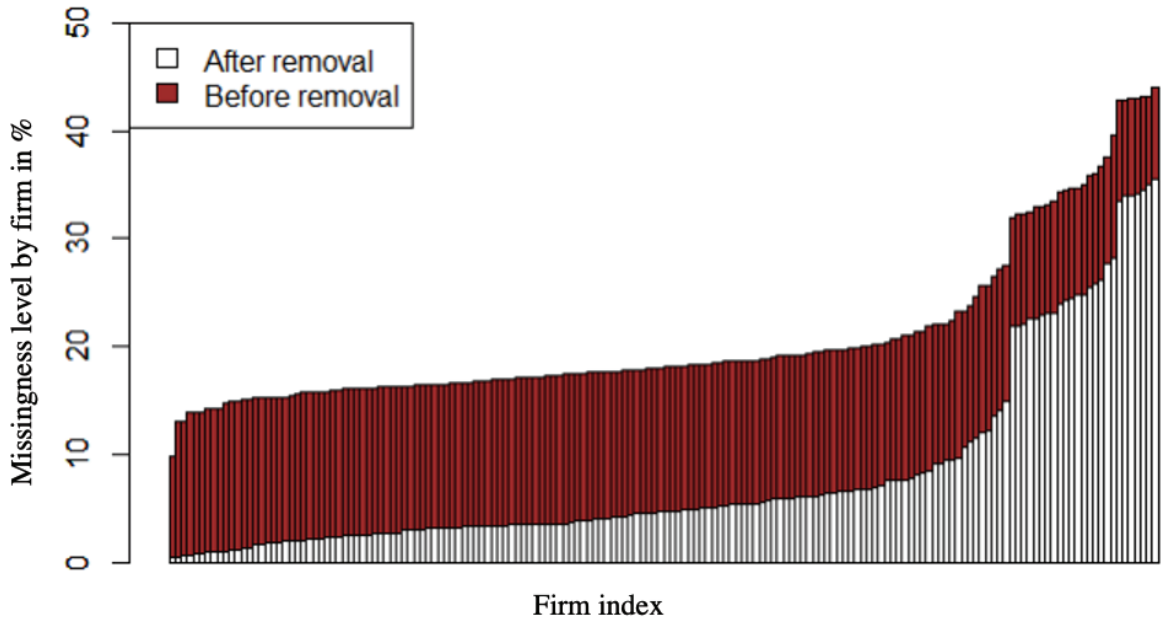
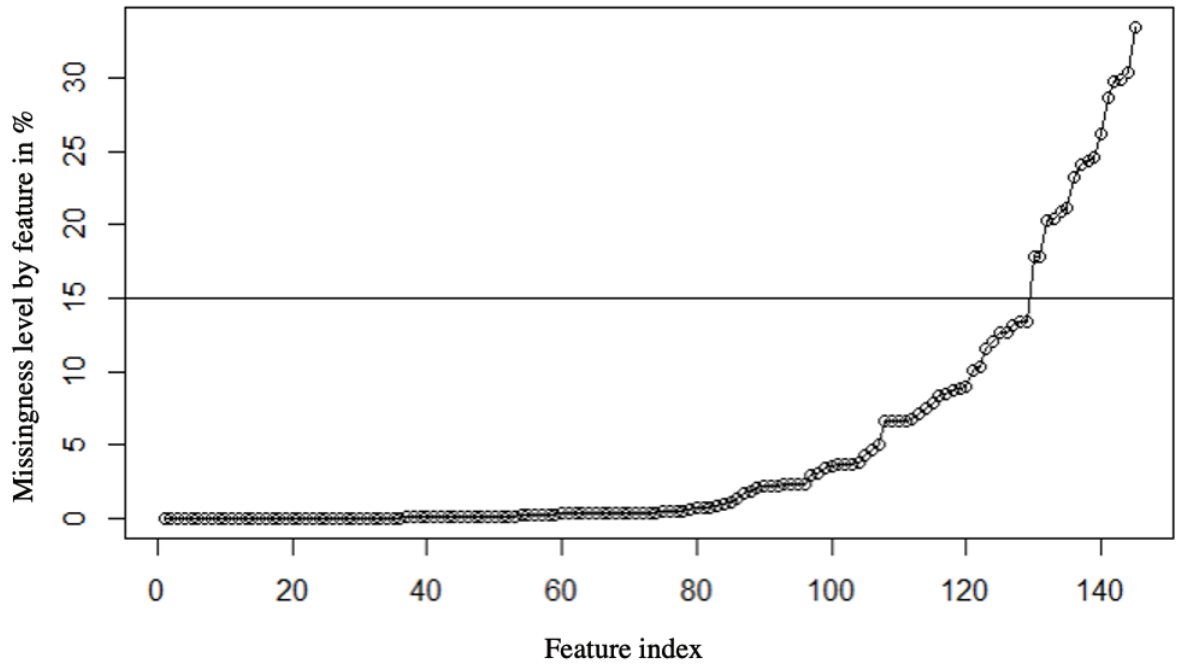


Figure 3: Bar graph showing the level of missing data by firm before and after removing the most missing features.



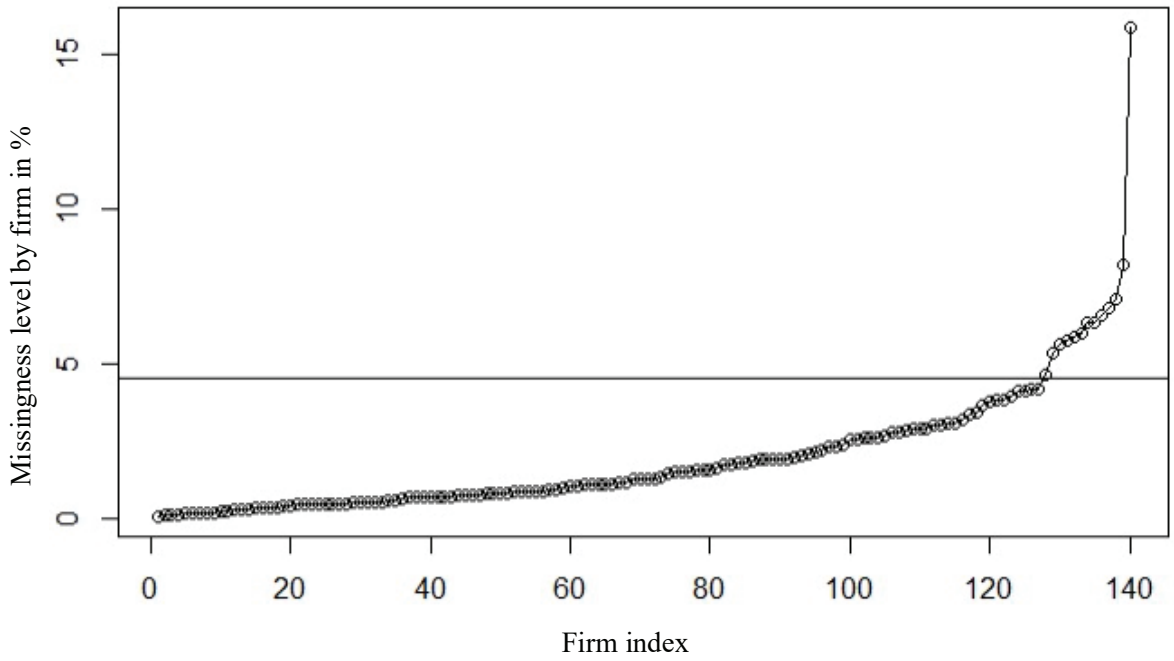
Despite achieving the overall missing rate of just under 4% for the entire dataset and of under 10% for all sectors, when analysing the missingness by variables or firms, there still were variables and firms with high missing rates, up to 33% missingness in a single variable and 21% in a single firm. Based on the literature, only variables identified as powerful predictors and with a missing rate of less than 15% were kept in the dataset. Some variables that were derivatives of other variables in the study and had high missing rates of more than 15% were also removed. Figure 4 shows the variables plotted against their missing rates. At the 15% mark there is a discontinuation separating variables that have a relatively much higher missing rate than the rest. A straight line on the 15% mark is plotted to show this discontinuity.

Figure 4: Line graph showing features and missingness level by feature, sorted by missingness level.



The number of variables left in the dataset was a hundred and twenty-three, which is a little higher than the studies that use only theoretical knowledge but much lower than the studies that use every possible variable available. The overall missing rate in the dataset was 1.6% which is low, however some firms still had relatively high missing rates. Figure 5 shows firms plotted against their missing rate. There is a continuous line of observations and a discontinuation at the 4.5% mark beyond which the firms have higher missing rates. This is shown in the plot with a horizontal line across at that point. Firms above the discontinuation were excluded. The final missing rate in the entire dataset was 1.4%, and beyond this point removing variables would result in removing some crucial variables as identified by the literature (Hajek and Michalak, 2013).

Figure 5: Graph showing firms and missingness level by firm, sorted by missingness level.



As explained in the introduction and literature review, there are twenty-one different credit rating intervals. Of these, seventeen are present in the dataset and are each assigned a number to use in the data analysis process. The different credit rating levels result in a dataset with imbalanced classes. Using literature as a guide, the classes are grouped such that the imbalance is either removed or reduced. Hwang, Chung and Chu (2010) split their classes into three equal groups. However, this study splits the seventeen ratings into four similarly weighted groups by using the S&P short term rating divisions as a guide. This division is shown by the solid line in between the rating levels in Table 4. S&P derives its short-term ratings from the long-term ratings. Splitting the ratings into classes helps to maintain the overall ranking of the S&P's credit rating scale. Table 4 also shows the equivalent short-term rating for each long-term rating. The bar plots in Figures 6 and 7 help visualise the distribution of credit ratings and their imbalance or skewness as well as the effect of grouping them.

Table 4: Table showing the percentages of each credit rating class before and after grouping.

Composition before grouping in %	Long term rating	Short term rating	Composition after grouping in %
0.87	AAA	A1+	27.42
1.22	AA+		
2.13	AA		
5.67	AA-		
6.22	A+	A1-	
11.32	A		
18.21	A-	A2	34.23
16.02	BBB+		
13.39	BBB	A3	21.20
7.81	BBB-		
5.83	BB+	B	17.15
4.13	BB		
4.55	BB-		
1.40	B+		
0.91	B		
0.20	B-		
0.14	CCC+	C	

Source: (S&P Global Ratings, 2021a)

Figure 6: Bar plot showing distribution of credit ratings before grouping.

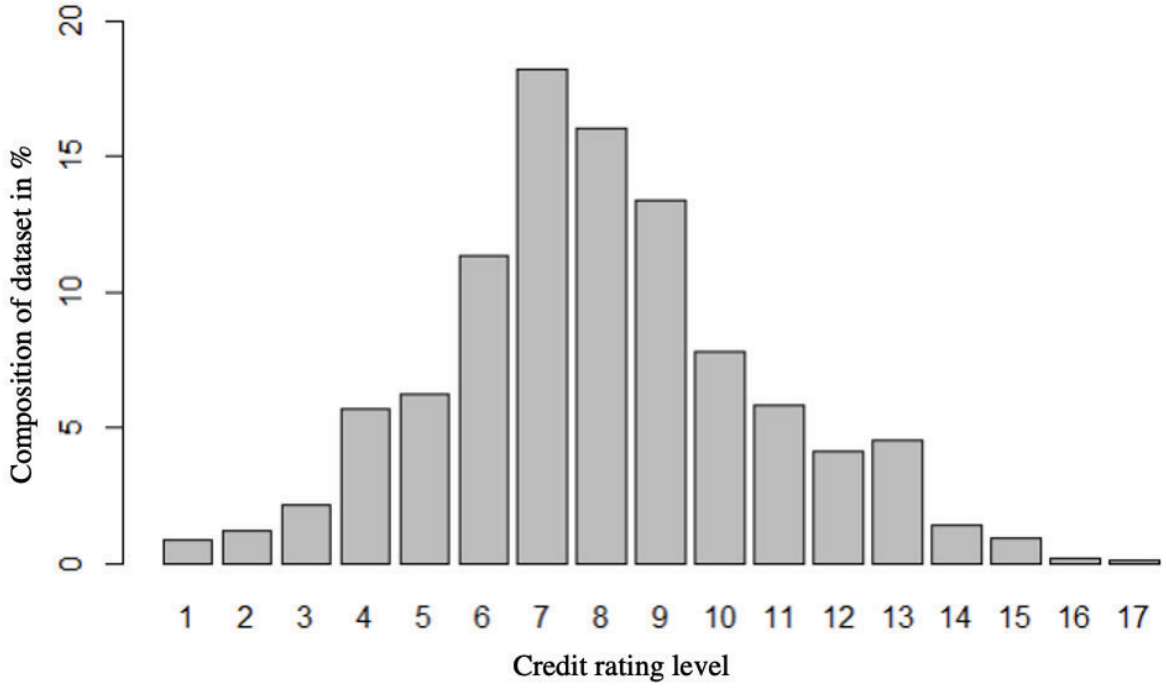
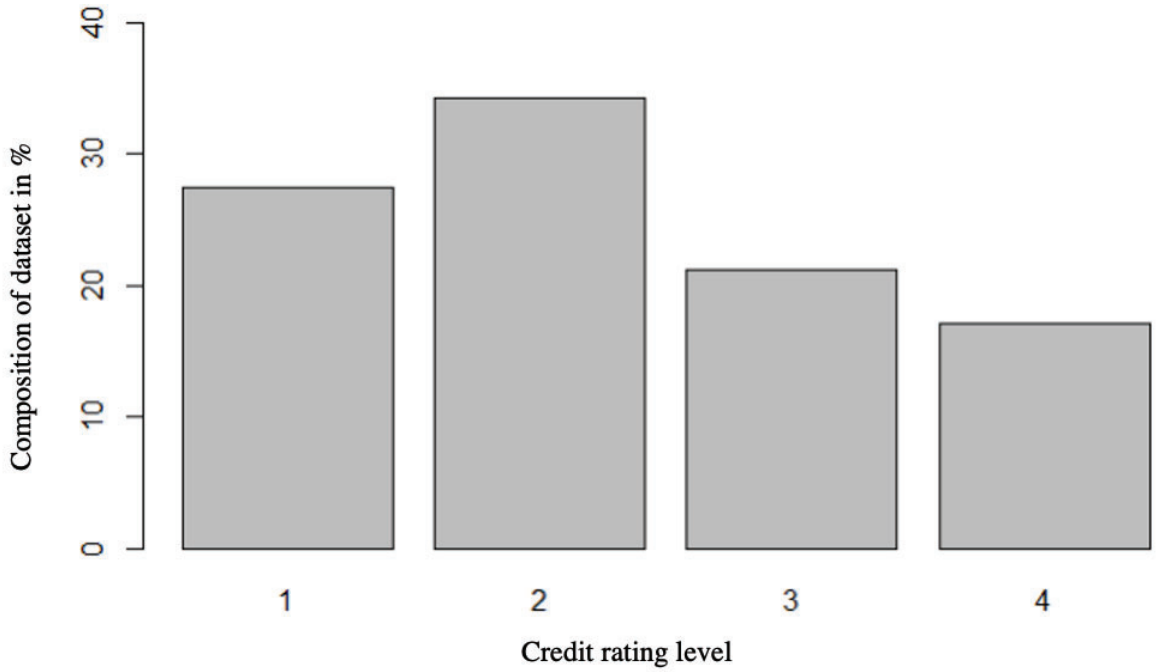


Figure 7: Bar plot showing distribution of credit ratings after grouping.



3.4 Handling of missing data

The final dataset has 127 firms with 123 explanatory variables and an overall missing rate of 1.4%. While this is a very low missing rate (Schafer, 1999), there were still some crucial variables that had missing points and removing more data beyond this point would reduce the sample size greatly without reducing the missing rates proportionately.

The approach used in this study to address the missing data problem was to impute the remaining missing data using multiple imputation by chained equations (MICE). While this method is not new, its application to credit rating datasets is. MICE was selected for three reasons. The first is its flexibility, this is twofold, first in using different algorithms to predict the missing data. The second is its ability to handle different types of data. The final reason is that MICE has been shown to be one of the best imputation methods across different studies (Ambler, Omar and Royston, 2007; Baneshi and Talei, 2012). To apply MICE, first, we need to identify patterns in missing data to classify the type of missingness (McKnight et al., 2007). For this a package in R created by Kirkegaard (2016) based on McKnight et al. (2007) was used to identify the type of missingness in the dataset.

A unique identifier for each pattern of missing data is tabulated as a percentage of the total cases of missing data, giving an estimate of the 'missingness complexity'. The missingness complexity simply tries to measure how many cases of missing data can be explained by patterns that exist within the data. If there are no patterns in the dataset, it can be assumed that the data is either MCAR or MNAR. In this case, however, there were several missingness patterns with a low missingness complexity implying that the data could be MAR. To strengthen this argument, a matrix is created that shows the relationship between a variable's missingness and its effect on another variable.

Table 5 is a snippet of the matrix. In this case, the matrix shows that there are indeed patterns in the missing data explained by the data. For example, it can be seen that variables with missing data such as dividends paid and gross margin, on average, have lower profit margins and return on capital. Thus, it can be assumed that the data is MAR and MICE can be used to fill in the missing data.

Table 5: Table showing relationship between variable's missingness and its effect on another variable.

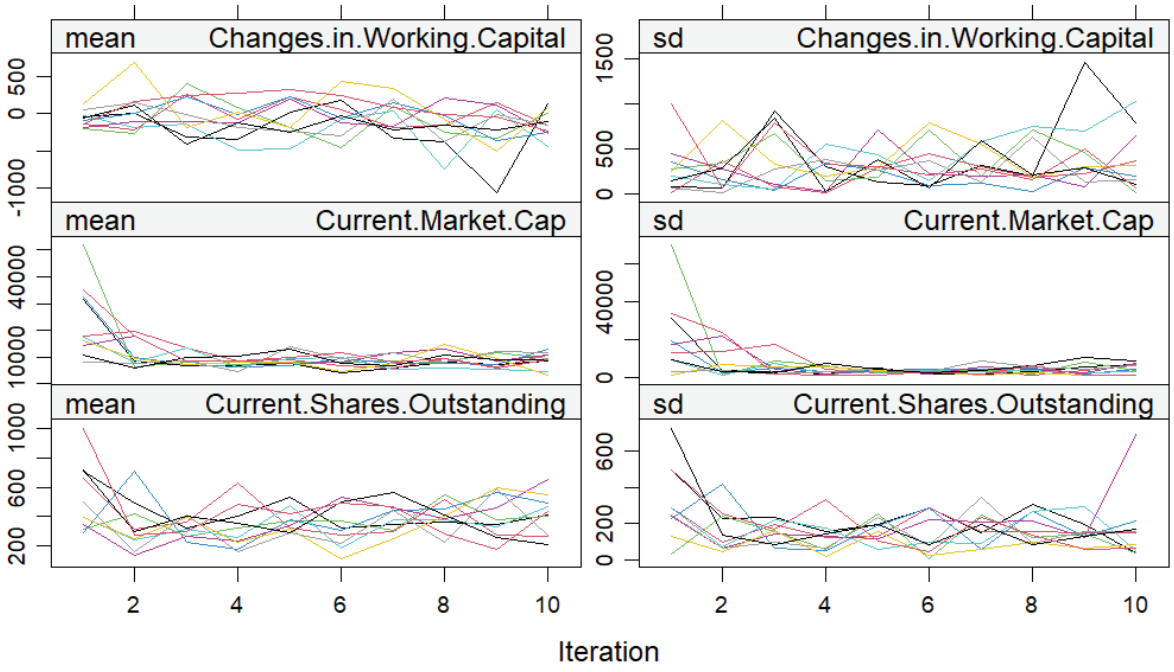
	Profit Margin	Return on Capital
Dividends Paid	-1.084	-0.882
Gross Margin	-0.429	-0.310

The process of multiple imputation by chained equations can briefly be explained as follows. Suppose we have a dataset with variables x_1 to x_n where some variables have missing values. If the first variable with missing values is x_1, then x_1 will be regressed against all the other variables in the dataset except for x_1. After the missing values in x_1 have been replaced with imputed values, the same process is repeated for all variables with missing values until there are no missing values in the dataset (Azur et al., 2011). This process is repeated several times to create multiple complete datasets until the results stabilise (Royston and White, 2011).

This study used the 'mice' package in R to perform multiple imputation by chained equations. The 'mice' package is explained in detail by Buuren and Groothuis-Oudshoorn (2010). The algorithm used to predict the imputed values was 'CART' or classification and regression trees. This is because its non-parametric nature makes it flexible for all types of data.

Ten iterations were used, and the algorithm took six hours to run. After the imputation, a brief analysis of convergence was carried out, as described by Buuren and Groothuis-Oudshoorn (2010). The analysis showed that almost all the imputed variables reached a steady state. A snippet of three variables' convergence graphs are shown in Figure 8. Since there are two graphs for each variable, not all of them can be shown and thus, they are linked in Link 2 of Appendix B. Having completed the imputation, the final dataset was completed with no missing values.

Figure 8: Image showing output plots from the MICE package in R that can be used to assess convergence of the imputation process.



Chapter 4
Models

4.1 Introduction

This chapter discusses the models used in this study, how they work, how they are trained and how their performance is evaluated. The models considered are random forests (RF) and long short-term memory neural networks (LSTM). We also discuss multilayer perceptrons (MLP) and recurrent neural networks (RNN) simply to provide context for the LSTMs.

4.2 Random forests

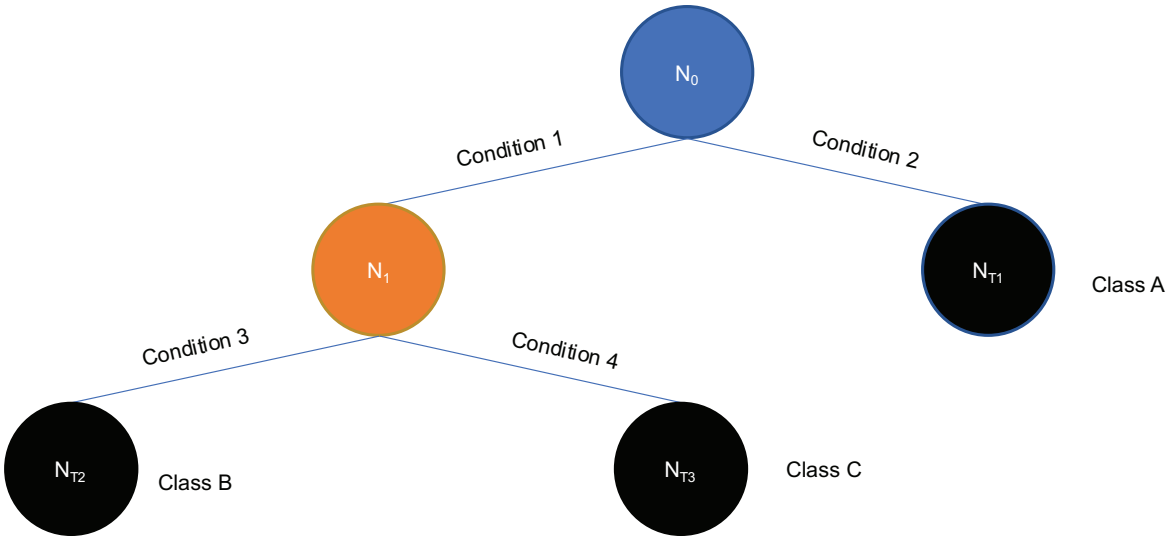
Random Forest models have been shown to be the best machine learning algorithm for predicting corporate credit ratings in several comparative studies (Golbayani, Florescu and Chatterjee, 2020). This section will describe the random forest algorithm, its assumptions and any pre-processing of the data that needs to be carried out prior to its application. This will be followed by the description of how the algorithm is trained and the optimal model is chosen.

Random forest is an ensemble model that is built on classification and regression trees. It attempts to retain the benefits of trees while mitigating against overfitting – a serious weakness with decision trees (Hastie et al., 2009). To describe random forests, a brief overview of classification and regression trees is provided in the next paragraph which puts into context the benefit of using random forests in this application.

Trees are predictive models that use the recursive binary splitting algorithm to partition the dataset in order to make predictions (Gareth et al., 2013). Trees are non-parametric, non-linear, and can deal with datasets that have numerical or categorical or a mixture of both variables without requiring any transformation nor do they require any assumptions about the dataset. Moreover, trees do not have any concept of time. They mimic human decision making and are highly interpretable (Gareth et al., 2013). However, trees suffer from high sampling variability. This means that tree models are not robust to different samples, as they tend to overfit a specific sample (Gareth et al., 2013). Figure 9 shows the architecture of a classification tree, where the

node N_0 is the root node and is the initial splitting criteria, the nodes that are in black are terminal nodes. These are nodes beyond which there are no other nodes (Gareth et al., 2013).

Figure 9: Diagram showing the architecture of a classification tree.



To solve the overfitting problem, several approaches have been proposed such as boosting, bootstrap aggregation and random subspaces. Boosting refers to a process of iteratively creating strong learners from weak learners (Zhou, 2012). Bootstrap aggregation, also known as bagging, is the process of creating an ensemble classifier by taking multiple samples with replacement from the dataset, training models on them and then aggregating the results (Breiman, 1996). The bagging process reduces the chances of overfitting a model. An improvement to the Bagging algorithm is the Random Forests algorithm.

The concept of random forests was formalised by Breiman (2001). Unlike bagging, Breiman's (2001) random forest samples the dataset as well as the feature space with replacement to create an ensemble of decision trees whose predictions are aggregated. The architecture of random forest involves creating multiple trees on different subsets of the data with different subsets of features. The number of features to be selected in each tree is generally referred to as 'mtries'. The results from all the trees are then averaged to a final result. Hastie et al. (2009) argue that although boosting is thought to have better performance than random forest, it depends on the dataset and application and that in general, random forest performs very similarly but is easier to train, tune and understand. Random forests can also be used to identify important features. This is obtained by calculating the difference in accuracy after the tree is split based on a certain

variable. The larger the improvement in accuracy when the tree splits by a certain variable, the more important the variable is considered (Hastie et al., 2009).

Generally, the random forest algorithm is used for both regression and classification. It requires very little pre-processing, and for most cleaned datasets it can be used directly. However, it is advisable to remove variables with high cardinality such as identity numbers or indices. On average two hundred trees are required for the performance of the algorithm to stabilise (Hastie et al., 2009). In addition to the number of trees, there are guidelines on how many features to select in each tree and the minimum number of data points to be used for prediction at each terminal node.

For classification, the guideline recommends the number of features to use to be equal to the square root of the number of explanatory variables, and the minimum number of data points to use for prediction in the terminal node to be 1; whereas for regression the corresponding numbers are a third of the explanatory variables and five respectively.

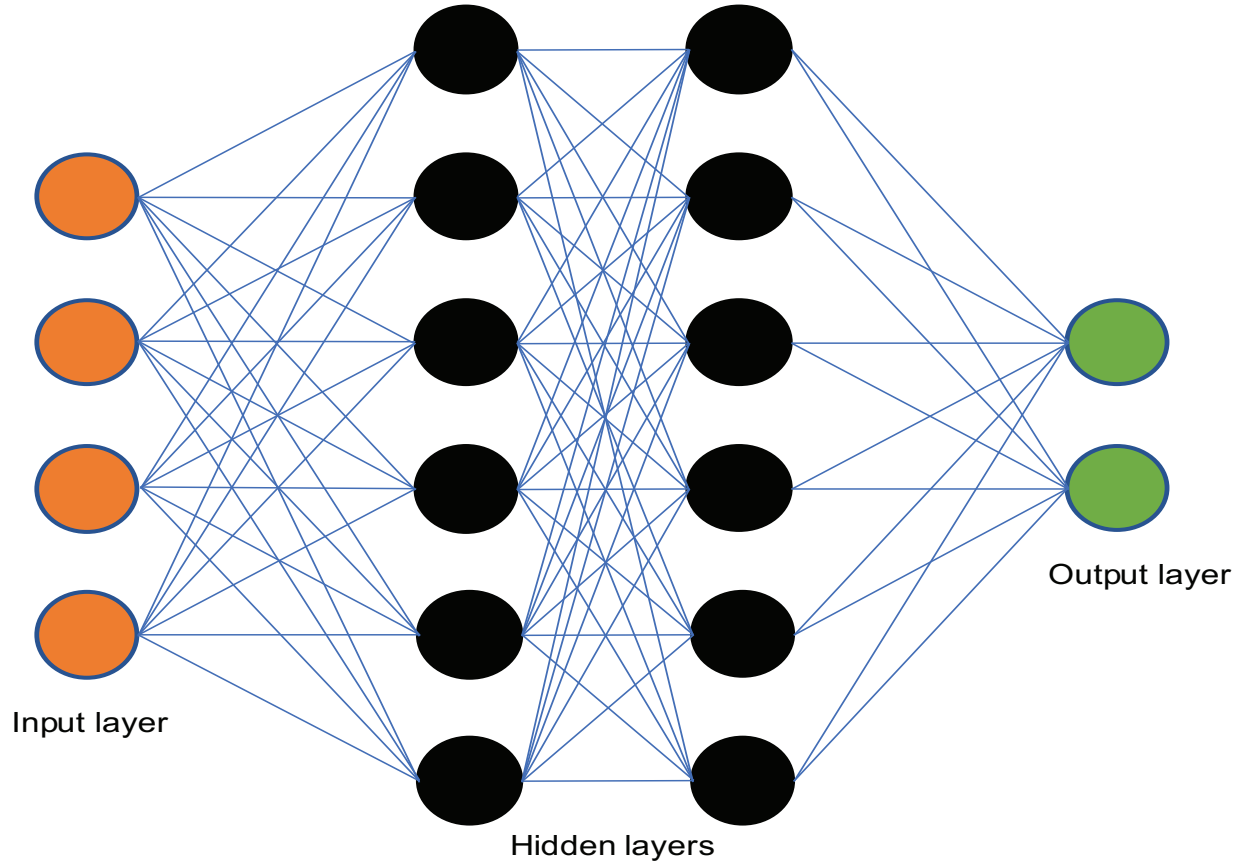
Cross validation can be done by using out of bag error which is similar to n-fold cross validation. Thus, in sequential problems, a random forest can be applied on the entire sequence. For regression problems, root mean square error and the mean squared error are generally used as accuracy metrics while misclassification error is used for classification problems. In both applications, sometimes log loss is used (Hastie et al., 2009).

4.3 Artificial neural networks

Artificial neural networks have risen in popularity and have been shown to predict accurately in many different applications, including financial uses such as credit rating predictions and stock price predictions. The simplest architecture of a neural network is a multilayer perceptron (MLP), where information flows one way only, from the input to the output and is processed one data point at a time.

The architecture of a neural network refers to the type of layers and number of layers it contains to form the neural network. There is an input layer to receive the sequence and an output layer to convert the nodes back to the required output. The layers that are in between the input and output are referred to as hidden layers. Neural networks are generally able to approximate any function with one sufficiently large layer, however this may be inefficient for some functions. Thus deeper layers are preferred as they generalize better (Reed and Marksll, 1999).

Figure 10: Diagram showing architecture of a multilayer perceptron.



4.4 Recurrent neural networks

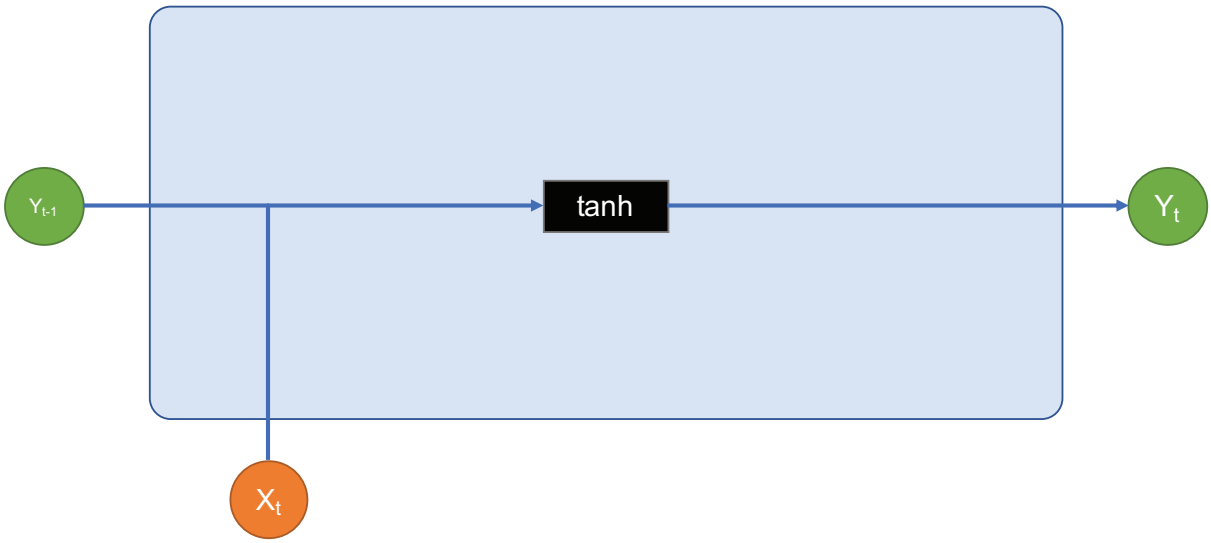
Multilayer perceptrons and similar feed forward NNs are poor at learning and predicting information that requires context. This is because each instance of a dataset is observed independently from the previous instance (Bengio, Simard and Frasconi, 1994). This is not always ideal for sequential information as it is possible that historic information has cues that are relevant to future predictions. For example, when predicting the final word in a sentence such as "BMW is a German car, I love the 2 series. It is my favourite. If I had to pick a country that I think makes the best cars, I would have to say it was Germany". In this sentence, the final word Germany can be identified as the correct word from the fourth word in the sentence.

A more relevant example would be predicting credit ratings for a company that does not sell consistently, such as a phone manufacturer that releases a flagship phone once every year or two. When predicting their credit rating, it would be worth considering their historic revenue or profit figures to determine their credit rating.

Recurrent neural networks (RNN) attempt to solve this problem. Their architecture differs from MLPs such that parts of it create feedback loops so that historic information can persist. This can be understood by thinking of the NN becoming a slightly different version of itself in every iteration of a sequence based on the information received on the previous input.

RNNs 'learn temporal data' using a technique called backpropagation through time (BTT). At every step the error between the input and output is calculated and through backpropagation the weights for the entire network are updated. Backpropagation is done by taking the partial derivative of an error at a specific step with respect to all the weights, and since this is sequential information, the number of timesteps at that point will be the number of derivatives required (Pascanu, Mikolov and Bengio, 2013). The use of the backpropagation method of learning forms the basis of RNNs' weakness in application. It leads to what is known as the vanishing gradient and the exploding gradient problems. This is when the weights (error signals) being sent as feedback while back propagating errors either get too large or too small (Pascanu, Mikolov and Bengio, 2013). In the case that the 'gradient explodes', the predictions will be inaccurate and the extent depends on the value of the weights. On the other hand, if the 'gradient vanishes', the RNN may take a very long time to learn or may fail to learn completely (Pascanu, Mikolov and Bengio, 2013). In practice, RNNs are shown to be unable to learn long sequences, and generally they will fail to learn a sequence with more than 5 timesteps (Gers, Schmidhuber and Cummins, 2000).

Figure 11: Diagram showing architecture of a recurrent neural network.



4.5 Long short-term memory neural networks

Long short-term memory neural networks (LSTM) avoid the vanishing gradient problem and therefore can learn long sequences and can be used in deep neural networks (Calin, 2020). They avoid the vanishing and exploding gradient problems by having a different structure to regular RNNs which engender a different method of BPT. Simple RNNs generally have a single layer while a typical LSTM unit has four layers that update the weights in a way that long term dependencies do not vanish (Hochreiter and Schmidhuber, 1997, cited in Olah, 2015).

LSTM units contain a cell state that is connected in the entire processing sequence. The cell state does not get transformed much apart from some linear transformations, which allows information to persist (Hochreiter and Schmidhuber, 1997). While LSTMs in practice are a remedy to both the vanishing and exploding gradient problems, theoretically, LSTMs can still have exploding gradients depending on the application (Calin, 2020).

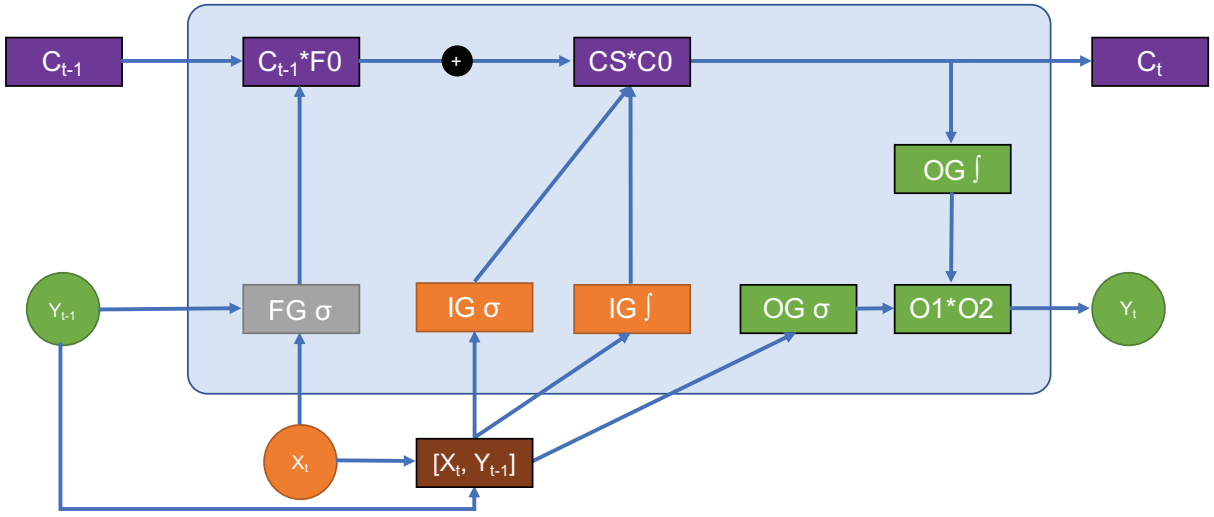
A typical LSTM unit has four layers that work in an additive way unlike regular RNNs. The first layer is called the 'forget gate' (FG), which is normally a sigmoid layer and outputs a vector F0. This layer determines the importance of the previous cell state (Ct-1) and what values should be kept or forgotten. A value of 1 means retain the corresponding value in the old cell state while a value of 0 means 'forget' that value in the old cell state and the value will be replaced with newer information. A number in between would scale down the values. The final step is to multiply the C0 with F0 to get the relevant remnants of the old cell state (Hochreiter and Schmidhuber, 1997; Olah, 2015).

The second layer is called the 'input gate' (IG). This layer has two components, the first component of this layer is a sigmoid layer that outputs a vector (CS) identifying which values of the current cell state should be updated. This is similar to FG and we get a set of values which are between 0 and 1. The second component is a hyperbolic-tangent layer that outputs a vector C0 with the potential values for the current cell state (Ct). C0 is multiplied with the vector CS and added to the remainder of the old cell state (Hochreiter and Schmidhuber, 1997; Olah, 2015).

In the third layer, the old cell state is updated, C0 is multiplied with the vector CS and added to the remainder of the old cell state thus preserving the relevant values from the old cell state and adding the relevant values from the new cell state.

The final layer is the 'output gate' (OG). Here the outputs are computed from the old cell state, the new cell state and the input in two steps. First a sigmoid layer processes the values from the old cell state, the input and the non-zero values of this matrix (O1) determines what parts of the cell state should be outputted. Then the current cell state is transformed in a hyperbolic-tangent layer giving a vector O2 of values within -1 and 1. Then O1 and O2 are multiplied to give the final output (Yt). (Hochreiter and Schmidhuber, 1997; Olah, 2015).

Figure 12: Diagram showing architecture of a long short-term memory neural network.



LSTM is generally applied to sequential and time series data such as text, speech, stock prices and other similar applications. Neural networks are very flexible, however, some broad guidelines to their use exist. In the case of LSTM the data fed into the network has to be a three dimensional sequence containing the samples, timesteps and the explanatory variables. The simplest method is to have a single hidden LSTM layer but in practice stacked LSTMs are used, where there are multiple hidden LSTM layers. LSTM layers can also be followed by dense and dropout layers. The basic optimizer used is 'Adam' and the mean squared error is used as a metric of accuracy in regression cases, in classification cases however, the misclassification rate is used (Brownlee, 2018).

Chapter 5
Application of models to data

5.1 Introduction

This chapter describes in detail the steps taken to fit the RF and LSTM models to the credit rating data for the purpose of using them to predict future credit ratings and comparing their performance. First, a description of the experiments that were used to address the objectives of the study as outlined in chapter one is given. Second, a description of any pre-processing of the data that was required before fitting each model and finally, the algorithms used in experiments are described in detail. As a reminder the objectives of the study are restated as follows. (1) To establish whether LSTM is better than RF at predicting credit ratings. (2) To determine if there is an optimal window size for predicting credit ratings, that is, is it better to use more recent data or is there benefit in having more historical data. (3) To establish whether preselecting features has any impact on the performance of LSTM models.

There are several experiments run in this study requiring a varied sequence length. The experiments can be further grouped into two broad categories, predicting 2 quarters into the future while varying the number of training quarters, and predicting the same number of quarters into the future as the number of training quarters. Two quarters was selected as the minimum size of a prediction window in line with the requirements of the United States Securities and Exchanges commission for companies to report data at the end of their quarters as discussed in Section 2.6. Table 6 shows a summarized list of the experiments conducted.

Table 6: Table showing a summarised list of experiments.

Experiment no.	No. of quarters input	No. of quarters to predict
1	2	2
2	4	4
3	6	6
4	8	8
5	10	10
6	4	2
7	6	2
8	8	2
9	10	2

Both the LSTM and random forest algorithms have specific requirements and nuances that must be taken into account when training them. However, there are three procedural aspects that are common to both methods during their application. The first is the method of splitting the train and test data in each experiment. The second is the use of the sliding window method to convert a time series dataset into a supervised learning problem. In the sliding window method, prior time steps, p, are used to predict one or more time steps, q, ahead. The number of time steps, p+q, is referred to as the window width or size in this thesis. This also allows the implementation of cross validation. The third, is the use of the grid search to identify the best model.

5.1.1 Data splitting into training and testing sets

The data is split in chronological order. The training set contains all the data except the number of quarters required to simulate a real life testing scenario. For example, if the time series data consists of T timesteps, and timesteps x(1), x(2),..., x(t) are used for training then the timesteps, x(t+1), x(t+2),...,x(T), are used for testing the prediction of the model once trained. Note that the data made up of the testing timesteps are never used for training, so that testing can be done

on an entirely unseen dataset. In this study a timestep refers to a quarter year. Table 7 shows the train and test split for each experiment defined in Table 6.

Table 7: Table showing the train and test split for each experiment.

Quarters	Exp 1	Exp 6	Exp 7, 2	Exp 8	Exp 3, 9	Exp 4	Exp 5
2	train	train	train	train	train	train	train
4	train	train	train	train	train	train	train
6	train	train	train	train	train	train	train
8	train	train	train	train	train	train	train
10	train	train	train	train	train	train	train
12	train	train	train	train	train	train	train
14	train	train	train	train	train	train	train
16	train	train	train	train	train	train	train
18	train	train	train	train	train	train	train
20	train	train	train	train	train	train	train
22	train	train	train	train	train	train	test
24	train	train	train	train	train	train	test
26	train	train	train	train	train	test	test
28	train	train	train	train	train	test	test
30	train	train	train	train	test	test	test
32	train	train	train	test	test	test	test
34	train	train	test	test	test	test	test
36	train	test	test	test	test	test	test
38	test	test	test	test	test	test	test
40	test	test	test	test	test	test	test

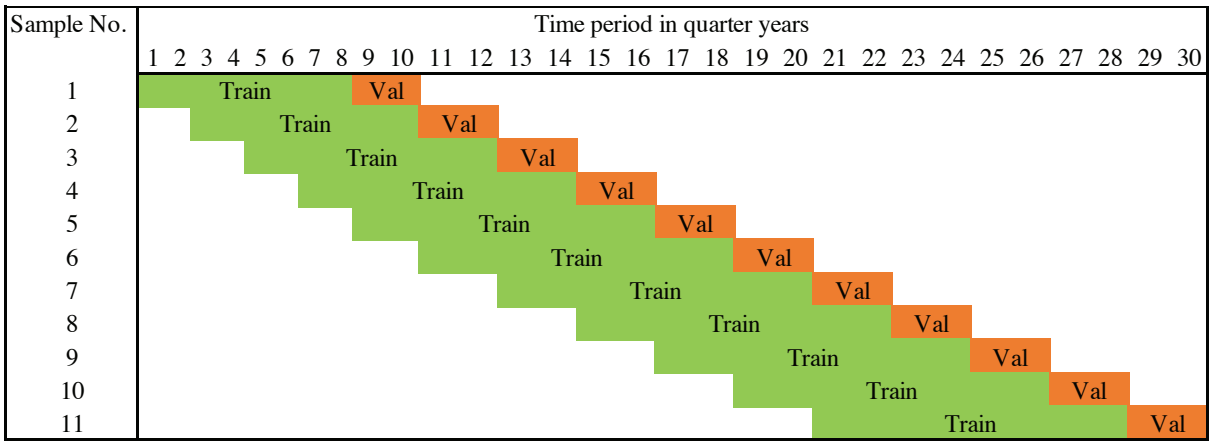
5.1.2 Cross validation

The method used for cross validation in this study is the sliding window method. To describe cross validation using the sliding window, let p and q be the number of prior quarters to train the model and the number of quarters for validating the predictions respectively, so that, w, defined as the size of the sliding window is given by p+q. The sequence made up of the p+q data points represent a sample from the training set. During training, the window slides over the training set starting with the first sample made up of the input sequence x(1). x(2),..., x(p) and the validation sequence x(p+1), x(p+2),..., x(p+q). At the next iteration, the window is shifted by an integer amount, s, known as the stride and a new sample is created. The stride is a hyperparameter selected by the user. The new sample created following the shifting of the window, consists of the input sequence x(1+s), x(2+s),..., x(p+s) and the validation sequence becomes x(p+1+s), x(p+2+s),..., x(p+q+s) respectively.

The process is repeated until the window reaches at most the last data point, x(t), in the training set. In general, if there are t training data points in a time series and the sliding window consists of input training timesteps of size p and validation or prediction timesteps of size q, then during iteration number, i, where i=0, 1, 2,.., (t-p-q)/s, of a training epoch, the input training and validation sequences will consist of the points x(1+i*s). x(2+i*s),..., x(p+i*s) and x(p+1+i*s), x(p+2+i*s),..., x(p+q+i*s) respectively. For this to work, (t-w)/s>=0. Clearly, the quality of the predictions will depend on t, p, q and s. The sliding window method best represents a real life scenario, since there is little to no benefit in predicting credit ratings of the past as they are known. The diagram in Figure 13 below shows an example of a sliding window used in this study in which t=30, s=2, p=8 and q=2. And this results in 11 iterations including the initial iteration when i=0.

This study applied sliding window cross validation for two reasons. The first was to avoid bias where the training data is in the future and the test data is in the past. The second was to determine if there was an optimal window size. This was based on the fact that many studies look at financial data and avoid major events such as financial crises. Such an experiment can help answer the question on whether it is better to use recent data since it better represents the current conditions of the market.

Figure 13: Diagram showing an example of sliding window cross with a window size of 10 quarters and a stride of 2 quarters.



Finally, grid searching is used to identify the best model. Grid searching involves defining a list of hyper parameters and running all the resulting models from all the possible combinations of those hyperparameters on the training set. For example, if there are two hyperparameters being tuned, namely, A and B, and if there are 5 candidate values for A and 10 candidate values for B, then a total of 50 models will be run.

5.2 Application of random forests to data

The performance of the random forests models in this study will be used as the benchmark for the expected performance of the LSTM models as random forest models have been shown to outperform other neural network based models. The first experiment run will be predicting two quarters into the future while varying the number of training quarters. In the second experiment, the number of quarters being predicted will be equal to the number of training quarters.

5.2.1 Data preparation for the random forest algorithm

The data used in the random forests consisted of the clean credit rating data from Chapter 3. The data was first split into training and testing sets and then converted into a dataset suitable for the application of supervised learning algorithms as mentioned in the introduction of this chapter. Two types of experiments are run, one predicting two quarters into the future and one predicting an equal number of quarters as used in the training process.

For the first set of experiments the number, p, of training quarters considered were: 2, 4, 6, 8 and 10 while the number, q, of quarters predicted was fixed at 2. In the second set of experiments, p was set equal to q where p, q are in the set, {4, 6, 8, 10}. That is, for the second

set of experiments, for each experiment, the number of training quarters used was the same as the number of predicted quarters.

Random forests are implemented in R using the H2O package. When training with random forests, high cardinality variables such as indexes tend to be overrated when building the model. To ensure the model generalizes, the high cardinality variables, specifically the ticker and time index are removed so that the model does not overly rely on them and generalizes sufficiently.

5.2.2 Training the random forest models and identifying the best models

All the random forest models were run in R with an H2O backend on a MacBook Pro M1 chip with 32 GB of RAM. The main hyperparameters that need to be specified to the random forest algorithm in H2O are the number of trees (ntrees), the number of variables (mtries), the minimum leaf size (min_rows) and tree depth (max_depth). H2O has built in functionality for grid searching through a list of hyperparameters to identify the best performing model. The hyperparameters used for running the random forest were as follows, mtries: 10, 25, 50, 75, ntrees: 50, 100, 200, min_rows: 1, 5, 10 and max_depth: 5,10, 20. The stride length for all random forest models was set to 2, this was to reduce the computational time and cost while still covering all the available quarters for validation.

H2O does not have a built in time based cross validations such as the sliding or expanding windows. However, it does allow the user to disable automatic cross validation and manually specify the training and validation fold. Thus, a simple for loop can be used in conjunction with the grid search function to feed all the data in a sliding window format. The for loop runs as follows: for each iteration, i, i=0, 1,..., (t-p-q)/s in an epoch, the model is trained using the p input quarters and predictions validated using the q validation quarters. At each iteration within an epoch, the training and validation is done for all possible hyperparameter combinations totalling 108, through the use of grid search.

The best model from each iteration is identified by its log loss, the model with the lowest log loss is defined as the best model in that iteration. This model is then stored along with all its details. In the next iteration the next sample is fed in and the process is repeated until the final sample. Refer to Figure 13 which shows a graphical representation of this process. Once the for loop is complete, a list of the hyperparameters from the best model in each iteration is

created. This list is then aggregated to create a consolidated model based on the information acquired from all the iterations, this serves two purposes, firstly and most importantly it allows the final model to be more generalized and representative of all the iterations. Secondly, it reduces the number of final models needed to be run on the test set.

The models are consolidated in three different ways resulting in three final models: 'min', 'avg' and 'max' models to be used on the holdout or test set. The 'min' model uses the minimum value of all the hyperparameters across the samples. For example, if an experiment has two samples, each sample will have a best performing model with its hyper parameters. If sample 1 has the number of trees as 50 and mtries as 10 and sample 2 has the number of trees as 20 and mtries as 50, the min model will have 20 trees and mtries set to 10. In case of the 'avg' models the average is taken while in the 'max' models the maximum values are used. The exception to this is when predicting ten quarters ahead using ten quarters of training data. There is only one model since there is only enough data to run one sample, thus the 'min', 'max' and 'avg' models are all the same.

5.3 Application of LSTM to data

This study used Keras with a TensorFlow backend in python to fit LSTM networks to the data. The LSTM layer in Keras is based on a model proposed by Hochreiter and Schmidhuber (1997). Unlike the H2O interface in R, Keras does not have in-built grid search functionality. Thus a simple grid search method was developed using basic Python functions and SciKitLearn's 'parametergrid' function.

5.3.1 Data preparation for LSTM

LSTM networks, unlike other feed forward NNs and other supervised learning algorithms, are said to have 'memory'. This means that the sequence in which training data is fed matters. Additionally, the LSTM in Keras used in this study, requires the timesteps of a sequence to be the same within a batch. This means that the output needs to be of the same length as the input thereby making it tricky to predict unequal sequence lengths.

There are solutions to this problem, the ones explored in this study included zero padding and sequence shifting. The first is to have the same length of output as the input and replace the observations that are not relevant with a zero. The second is to shift the output forward by the number of quarters that need to be tested, so that only required observations are novel. These

adaptations affect the training misclassification rate by either skewing it upward or downwards. Thus, the training misclassification rate are not directly comparable. When training with the zero padded sequence the zeroes are entirely ignored in the calculation; this skews the training accuracy downwards but is the more cautious approach. However, when training with the shifted sequences, a part of the target variable is not novel and the model is expected to predict these values with ease. Table 8 shows these sequences.

Table 8: Table showing the types of sequences used when modelling with LSTM.

Normal Sequence		**Shifted Sequence**		**Padded sequence**	
Train	**Predict**	**Train**	**Predict**	**Train**	**Predict**
1	5	1	3	1	0
2	6	2	4	2	0
3	7	3	5	3	5
4	8	4	6	4	6

When calculating the test misclassification rate however, only the quarters of interest are compared and used in the calculation to get the models' misclassification rate. Thus, all the test misclassification rates are directly comparable. Applying such a calculation in every batch of every epoch during training is very computationally expensive and time consuming and would be unfeasible in the scope of this study.

The dataset used in this study poses another challenge as it is a multiple time series dataset. In a single time series dataset, there is only one way to feed the data to the LSTM model. However, with a multiple time series dataset there can be multiple ways to feed the data while still maintaining chronological order. In our dataset there are forty time periods and each time period has data for all the firms. The fact that there are multiple examples (firms) at each time period, means that there can be two ways to feed the data. The order of feeding the data does matter when training LSTMs, unlike when training the random forest algorithm which lacks a temporal element.

The first approach of feeding the data to the LSTM model is to treat each firm's data as a separate time series and train the network one firm at a time. In this approach the cell state captures purely the temporal effect for a single firm (firm ordered). The second approach is to put all the firm's observations in each time period (time ordered). In this case the cell state captures a mixture of the temporal effect and the relationship between the firms. The difference between the two sequences are shown in Tables 9 and 10 where a sample of time ordered and

firm ordered data are shown respectively. The time ordered and firm ordered data might contain the same information but the difference in format could result in a difference in performance as the length of the sequences for firm ordered data are significantly shorter.

Table 9: Table showing a sample of time ordered data.

Index	Firm	Time	Rating
1	A	1	1
2	B	1	2
3	C	1	1
4	D	1	1
5	A	2	2
6	B	2	1
7	C	2	2
8	D	2	1

Table 10: Table showing a sample of firm ordered data.

Index	Firm	Time	Rating
1	A	1	1
2	A	2	2
3	B	1	2
4	B	2	1
5	C	1	1
6	C	2	2
7	D	1	1
8	D	2	1

For example, when training on 2 quarters a time ordered dataset would have a sequence length of 254 while a firm ordered sequence would only have a length of 2. In the first case there is a longer sequence but contains fewer batches per sample, the opposite is true for the second case.

5.3.2 Training the LSTM models and identifying the best models

The LSTM models in this study were run on Google Collaboratory Pro Plus on an NVidia Tesla T4 or an NVidia A100 GPU to reduce training times. The training of the LSTM models involved three steps. The first step was to standardize the data using a StandardScaler. The second step was to create the appropriate sequences and the final step was to feed those sequences into the LSTM models.

The LSTM neural networks in this study had an LSTM layer as the first hidden layer and a softmax output layer that converted the results into the required format. The LSTMs varied the number of hidden layers from two to four, and also varied the type of layers, combining LSTM layers with dense layers. Five distinct architectures were used, these are: (1) two stacked LSTM layers, (2) three stacked LSTM layers, (3) four stacked LSTM layers, (4) two stacked LSTM layers and one dense layer and (5) two stacked LSTM layers and two dense layers. A grid was created for each architecture, varying the following hyperparameters: the number of nodes in each layer, the type of activation function used in the dense layers, the optimizer and the dropout rate. The nodes in each layer were varied from 10 to 123. The dropout values were varied as follows: {02, 0.4, 0.6}. The activation function for the dense layers was varied between the hyperbolic-tangent function and the rectified linear unit function. The stride length for all LSTM experiments was set to 1 and the batch size 16, this was to increase the number of samples available for the algorithm to learn from.

When predicting with LSTMs using Keras, the input and output sequence need to be of the same dimensions. Thus, when trying to predict only two quarters while varying the number of training quarters, the data needs to be pre-processed. When fitting LSTM models, three different sequences were used, The first used the same length of quarters to train and predict. The second used zero padding in the output to net two quarters. The third used a sequence that was shifted forward by two quarters so that only the latest two quarters contained unseen data. Refer to Table 8 for an example of these sequences.

This meant that the training metrics were not comparable. To ensure that metrics for predicting the test set were comparable, calculations were carried with a custom function that only compared the quarters of interest. The calculations are computationally too costly to run in every epoch of training for several thousands of models. Despite the cost, in the case of the

testing data, it was found necessary to carry out the calculation to ensure that all the final results could be compared to each other.

As an additional experiment, this study investigates the performance of the model when the time ordered approach is used. In this case, the option with the minimum number of quarters, that is, 2 quarters will be studied. This is to confirm whether having a longer sequence and thus, more timesteps improve the performance of LSTM networks given that LSTMs are generally known to perform better with longer sequences (Brownlee, 2018). These sequence structures are discussed in Section 5.3.1. The time ordered sequences are trained the same way as the other LSTM models but with smaller grids for the sake of time and to avoid computational cost.

In all cases, the largest grid was run first, without any feature selection. The top ten ranked models (ordered by misclassification rate) from each quarter and sequence length are considered when defining the grid for the feature selection runs, this smaller grid was rerun with preselected features. The preselected features were obtained by fitting a random forest model on the training data for each experiment and obtaining the variable importances from the output. This would result in a list of important features for every unique sequence length. In this case there are seven unique sequences, see Table 7 showing the train and test split for each experiment

Finally, all the results are combined from the runs with and without pre-selected features. The top ranking model from each sequence length and type are then used for testing. This means that each sequence length and type would have at least one model. For example, the sequence that uses four quarters as training to predict two quarters ahead there will produce at least two best models, one for the unequal sequence and one for the shifted sequence.

Chapter 6
Results

6.1 Introduction

This chapter presents the results of the experiments that were described in the previous chapter. The results of predicting the credit rating using the random forest algorithm will be presented first followed by those in which the LSTM algorithm was used.

6.2 Results for random forest

The result from the experiments conducted with the random forest algorithm can be better reported by separating them based on the number of quarters predicted ahead. The first set of results relate to when the algorithm was used to predict credit ratings two quarters into the future while varying the number of training quarters; and the second set of results relate to when the algorithm was used to predict credit ratings for a number of quarters (more than two) into the future that were equal to the number of training quarters.

6.2.1 Results when predicting two quarters into the future with varying number of training quarters

Table 11, summarises the results of random forest when predicting 2 quarters into the future. The test misclassification rate is similar when the training quarters are varied from 2 to 10. The best performing model was MIN62. The model used six quarters of training data and achieved a misclassification rate of 3.5%. The next best misclassification rate was 4.7% and this was achieved by six models as shown in the Table 11. It should be noted that the model, MIN22, used only two quarters of training data to score this misclassification rate while the other five models used more than two quarters of training data. The worst performing model was MAX22, which used 2 quarters of training data and had a misclassification rate of 6.7%. Overall, the ‘min’ models outperformed the ‘avg’ and the ‘max’ models by a small margin.

Table 11: Table showing best performing RF models when predicting 2 quarters ahead on test data.

Training qtrs.	Predicting qtrs.	No. of trees	Max depth	Mtries	Model	Misclassification error(%)
2	2	50	10	25	MIN22	4,72%
2	2	132	14	49	AVG22	5,51%
2	2	200	18	75	MAX22	6,69%
4	2	50	15	10	MIN42	5,91%
4	2	103	17	36	AVG42	4,72%
4	2	200	20	75	MAX42	5,91%
6	2	50	10	10	MIN62	3,54%
6	2	123	18	39	AVG62	4,72%
6	2	200	20	75	MAX62	4,72%
8	2	50	17	10	MIN82	4,72%
8	2	105	20	40	AVG82	6,30%
8	2	200	20	75	MAX82	4,72%
10	2	50	10	25	MIN102	5,51%
10	2	89	19	44	AVG102	6,30%
10	2	200	20	75	MAX102	5,51%

6.2.2 Results when predicting equal numbers of quarters into the future as the number of quarters forming the training data

Table 12 provides a summary when the number of training quarters is equal to the number of predicted quarters. There appears to be a negative relationship between the number of quarters being predicted and the performance of the model, so that the more quarters there are in the prediction period, the lower the accuracy and the higher the misclassification rate. With the larger sequences - involving more quarters - the best performing models were the ‘avg’ models that slightly outperformed the ‘min’ and ‘max’ models. This observation is in sharp contrast to the results of predicting 2 quarters in the previous section. There is only one model for predicting ten quarters using ten quarters of training data as there is only one-fold given the method of the split. This is because the dataset is made up of 40 quarters and 20 quarters are

used for training while the other 20 quarters are used for testing. In this case, the 'min', 'max' and 'avg' models are the same.

Table 12 shows that the best performing model is MIN44 with a misclassification rate of 6.1%, an exception to the general trend where the 'avg' models are the best performing models. The MIN44 model has the largest number of trees among the 'min' models which compares closely to the average number of trees among the 'avg' models. The model with the highest misclassification rate of 20.6% is RF1010, which matches the general trend where, the number of quarters being predicted increases proportionally with misclassification rate.

Table 12: Table showing best performing RF models when predicting equal quarters ahead on test data.

Training qtrs.	Predicting qtrs.	No. of trees	Max depth	Mtries	Model	Misclassification error(%)
4	4	100	15	10	MIN44	6.10%
4	4	146	18	33	AVG44	6.69%
4	4	200	20	75	MAX44	8.46%
6	6	50	10	25	MIN66	13.78%
6	6	139	16	33	AVG66	12.20%
6	6	200	20	50	MAX66	12.07%
8	8	50	20	10	MIN88	17.62%
8	8	170	20	24	AVG88	17.13%
8	8	200	20	50	MAX88	17.72%
10	10	200	20	25	RF1010	20.63%

6.3 Results for LSTM

As with the results for the random forest, the results from the experiments conducted using the LSTM algorithm can also be better reported by looking at the experiments conducted on predicting two quarters separately from those conducted on predicting an equal number (more than two) of quarters into the future as there were training quarters.

6.3.1 Results when predicting two quarters into the future while varying number of quarters for training data

Table 13 is a summary of results for LSTM models with the normal, shifted and unequal sequences indicated as FO, FOSH and FOUE respectively when predicting two quarters into the future while varying the quarters of training data.

The best performing model when predicting 2 quarters ahead had a padded sequence, where 4 quarters were used for training and 2 for prediction. The overall best performing model was FOUE1 and used only 10 of the preselected features and had a misclassification rate of 32.28%. The model was closely followed by the model FOSH1 which also used 10 features and a shifted sequence with 4 quarters of data to predict two quarters ahead. The model had a misclassification rate of 32.68%. These models differ in performance in terms of misclassification error by less than half a percent and both use 4 quarters of training data to predict two quarters ahead.

Table 13 shows that the next best model was FO1 which used a normal sequence of two quarters of training data to predict two quarters ahead. The model had a misclassification rate of 34.65%, very similar performance to that of FOUE1 and FOSH1. The rest of the models consisting of longer sequences of training data to predict two quarters ahead did not perform any better. In fact, as the sequences got longer, there was a slight deterioration in performance.

The top three models all used the top ten most important features, and all shared the same neural network architecture consisting of two LSTM layers, followed by two dense layers; the same architecture used by eight of the best nine models. Additionally, only one of the nine best models contained all the features from the dataset instead of preselected features.

Table 13: Table showing the best performing LSTM models for firm ordered data on test dataset when predicting 2 quarters ahead.

Model	Architecture	Training qtrs.	Predicting qtrs.	No. Features	Validation Misclassification (%)	Test Misclassification (%)
FO1	LLDDO	2	2	10	30.21	34.65
FOSH1	LLDDO	4	2	10	29.88	32.68
FOSH2	LLDDO	6	2	10	28.11	41.34
FOSH3	LLDDO	8	2	25	29.82	39.37
FOSH4	LLDDO	10	2	123	31.55	37.40
FOUE1	LLDDO	4	2	10	39.94	32.28
FOUE2	LLDDO	6	2	10	38.02	37.01
FOUE3	LLDO	8	2	10	34.31	42.13
FOUE4	LLDDO	10	2	25	35.66	38.19

6.3.2 Results when predicting equal numbers of quarters into the future as the number of quarters forming the training data

The next set of experiments focused on using LSTMs to predict the same number of sequences into the future as the number of sequences for training data. In this case, no special pre-processing of the sequences is required. Similar to the results from the random forest, it was observed that as the number of quarters being predicted increased the performance of the algorithm decreased too. However, unlike in the random forests case, the misclassification rates did not change as drastically when the sequence lengths were increased. The results of the experiments are shown in Table 14.

Table 14 shows that the best performing model was FO4 which used eight quarters to predict eight quarters ahead. The model utilised twenty five preselected features and achieved a misclassification rate of 39.47%. Model FO4 was closely followed by model FO2 which used four quarters of training data to predict four quarters ahead. The model used only ten preselected features and achieved a misclassification rate of 39.57%, which is only 0.1 percent behind the best model.

Three of the top four models used the same architecture consisting of two LSTM layers followed by two dense layers. The model FO5 was the only one that did not contain a dense layer. Moreover, FO5 was used to predict the longest sequence where ten quarters of training data were used to predict ten quarters ahead. The model, similar to random forests, yielded the highest misclassification rate.

Table 14: Table showing the best performing LSTM models for firm ordered data when predicting equal quarters ahead on test data.

Model	Architecture	Training qtrs.	Predicting qtrs.	No. Features	Validation Misclassification (%)	Test Misclassification (%)
FO2	LLDDO	4	4	10	29.76	39.57
FO3	LLDDO	6	6	50	31.48	41.21
FO4	LLDDO	8	8	25	27.62	39.47
FO5	LLO	10	10	123	32.69	47.80

6.3.3 Results when sequences were fed to an LSTM network in time ordered structure

To identify what models to run in the time ordered sequence, the top fifty models were selected from the experiments conducted when using two quarters of training data to predict two quarters ahead. The top fifty models were used to create the grids to run in the time ordered sequence, These grids were run on the training set. The top three models from this training set were then run on the test set.

The results of the top three models when training data was fed to the LSTM using the time ordered approach are shown in Table 15. The models have a very low validation misclassification rate. However, their test misclassification rate is much higher. All three of the models performed accurately achieving misclassification rates of less than ten percent; which was better than any of the earlier models that took in data using the firm ordered approach. The performance of the time ordered based models on the test set was only slightly better than that of the firm ordered based models. The best performing model in the time ordered sequence had a misclassification rate of 28.35%, an improvement of 6.3% over the FO1 model encountered above.

Table 15: Table showing the best performing LSTM models for time-ordered data on test data.

Model	Architecture	Training qtrs.	Predicting qtrs.	No. Features	Validation Misclassification (%)	Test Misclassification (%)
TO1	LLDO	2	2	25	9.55	29.53
TO2	LLDDO	2	2	25	9.92	31.10
TO3	LLDDO	2	2	25	9.97	28.35

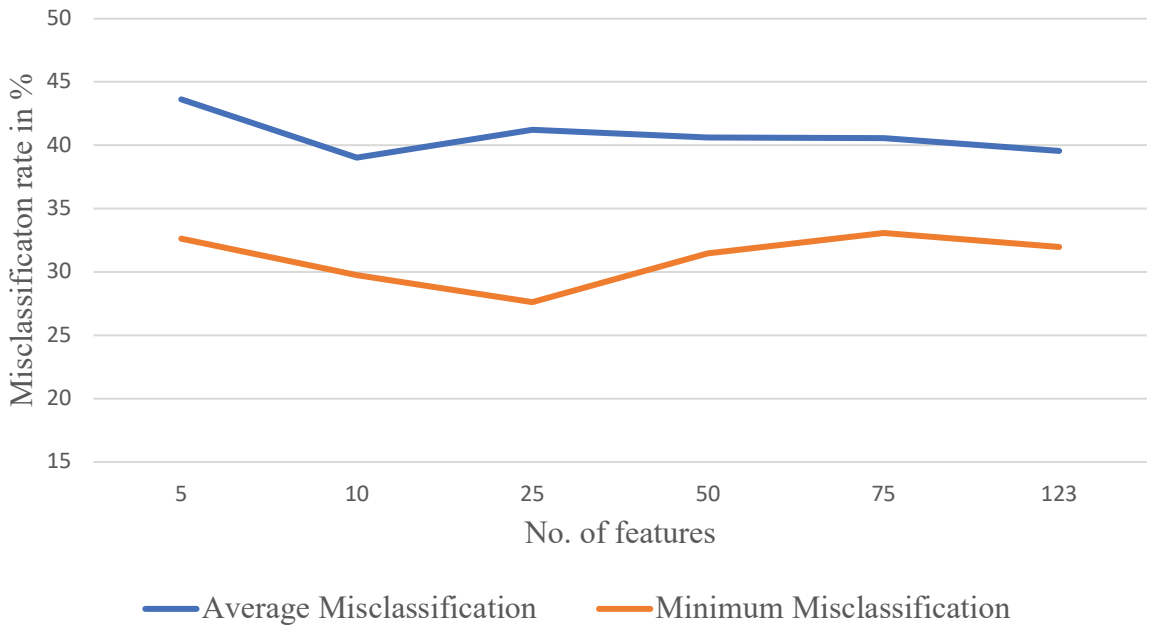
6.3.4 Impact of feature selection on LSTM performance

The experiments performed with LSTM neural networks involved using pre-selected features to observe its impact on performance. Only the training results for the normal sequences are directly comparable and so only those results are presented here. The line graph in Figure 14 shows the average and minimum misclassification rates plotted against the number of preselected features. The value 123 in the features axis represents no feature selection as all the features were used.

Figure 14 shows that there is a small difference in misclassification rate when feature selection is applied. The average and minimum misclassification rates are highest when the number of features is 5. The minimum misclassification rate is at its lowest when the number of features is ten and the average misclassification rate is lowest with 25 features. The performance the model appears to be stable beyond this point.

The testing results which contain the best models from all sequences also contain a majority of 10 preselected features followed by 25, with only two of the best models employing no feature selection and one model with fifty features. None of the best models had 75 pre-selected features. See Tables 13, 14 and 15.

Figure 14: Line graph showing the validation misclassification error for LSTM models on normal sequences when the number of features are varied.



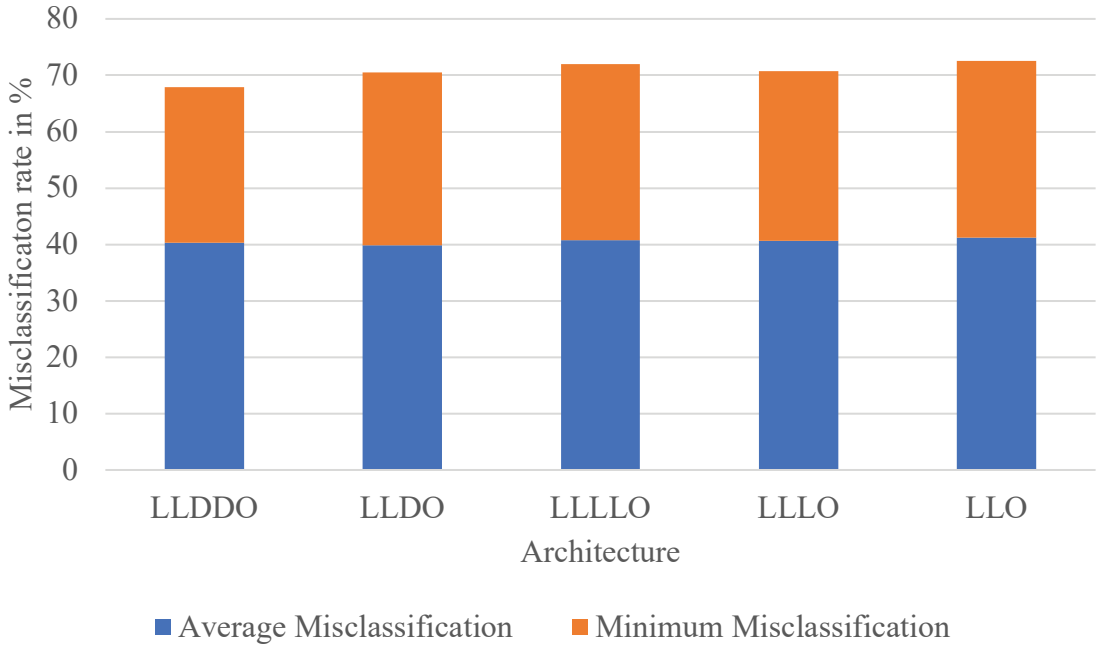
6.3.5 Impact of architecture selection on LSTM performance

The experiments performed using LSTM neural networks also varied the architecture of the neural networks, using LSTM layers only in some and combining LSTM layers with dense layers in others. The results for the impact of architecture can be summarized using a stacked bar graph shown in Figure 15. The stacked bar graph shows minimum misclassification rate stacked on top of the average misclassification rate. Similar to the impact of features, these results only considered the normal sequences as their training results are directly comparable.

It can be seen that the misclassification rates are very similar with all the results being within 3 percent of each other. The lowest average misclassification rate was obtained when the architecture consisted of two LSTM layers and one dense layer, while the lowest minimum and overall misclassification rate was obtained when the architecture consisted of two LSTM layers and two dense layers. The latter architecture was also the deepest neural network considered in the experiments. The highest average and minimum misclassification rates were obtained when the architecture consisted of two LSTM layers, this was the shallowest neural network considered.

These results can be corroborated by the testing results - even in the shifted, padded and time ordered sequences - as all the best models except for three, had the same architecture of two LSTM layers followed by two dense layers, see Tables 13, 14 and 15.

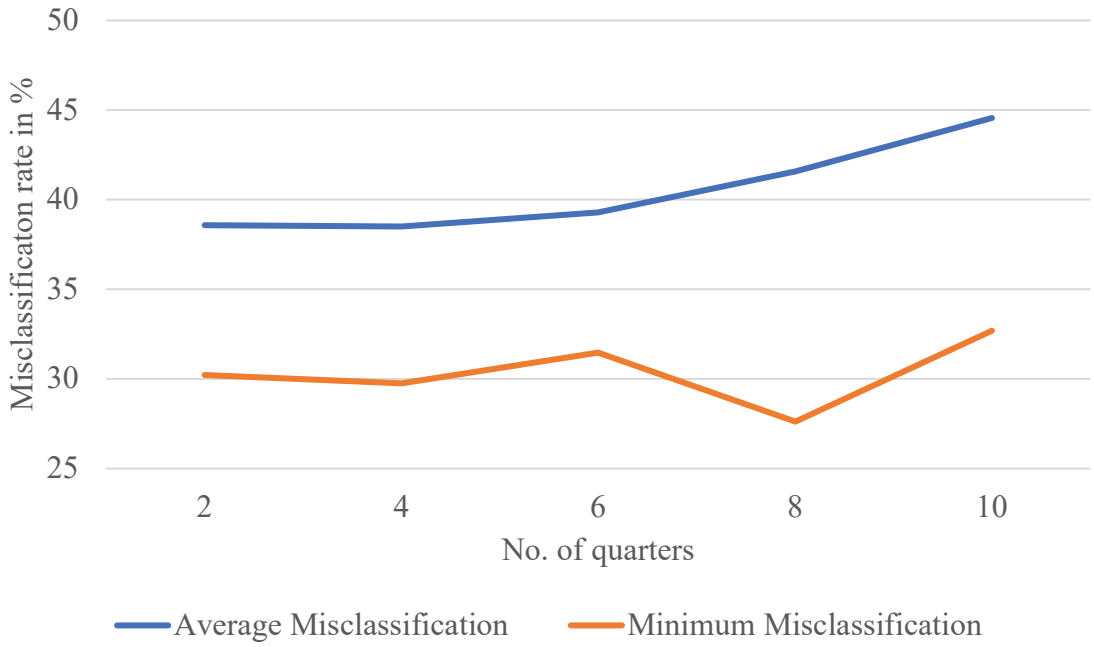
Figure 15: Stacked bar graph showing the sum of average and minimum validation misclassification error when the neural network architecture is varied for LSTM models on normal sequences.



6.3.6 Impact of sequence length for training data and prediction on LSTM performance

The sequence length, that is, the number of quarters being used for training and prediction were also varied. The results presented only consider the normal sequences as their training results are directly comparable. Figure 16 shows a line graph of the average and minimum misclassification rates against the number of quarters. The minimum misclassification rate is highest at 10 quarters and lowest at 8 quarters. The average misclassification rate shows a clear pattern, where the lowest average misclassification rates corresponds to the shorter sequences. The testing results show a similar pattern (See Table 14).

Figure 16: Graph showing the validation misclassification error when the number of quarters are varied for LSTM on normal sequences.



6.3.7 Learning curves from LSTM models

This section of the results presents the learning curves obtained from the best performing models for each sequence type in the LSTM experiments. These graphs were obtained by observing the training process over fifty epochs. Figures 17, 18, 19 and 20 correspond to the normal, padded, shifted and time ordered sequence respectively. All the learning curves follow a similar pattern. In the beginning, the training and validation curves mingle slightly and shortly after, the training curve drops quickly while the validation curve remains unstable until towards the end of the training period when it stabilises.

Figure 17: Learning curve for the best performing model on normal sequences.

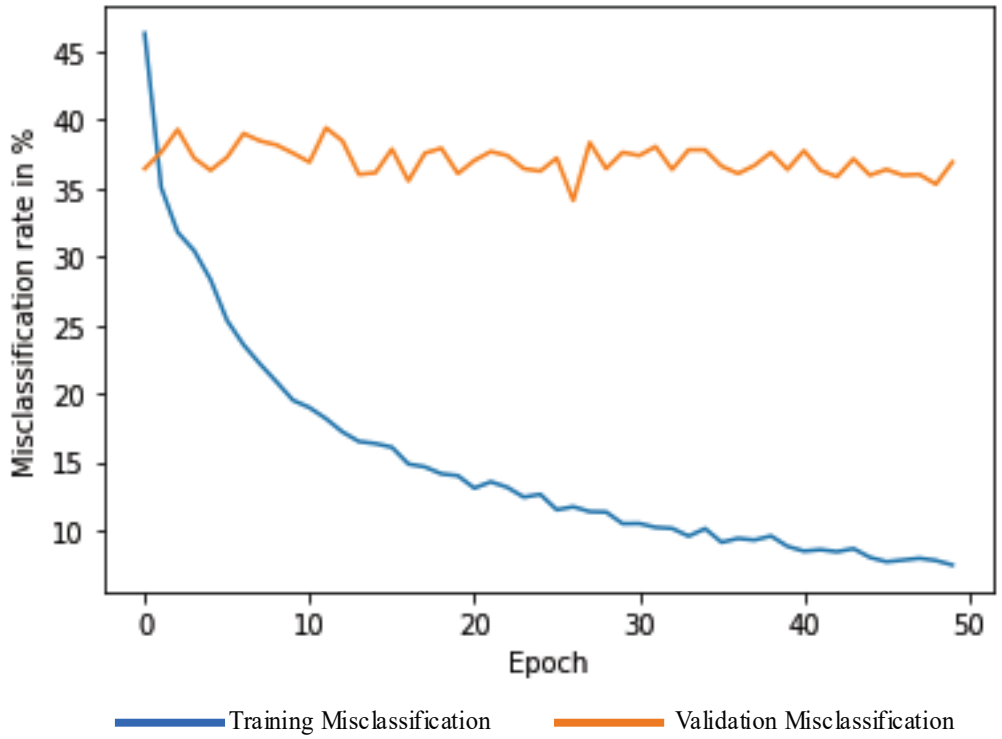


Figure 18: Learning curve for the best performing model on padded sequences.

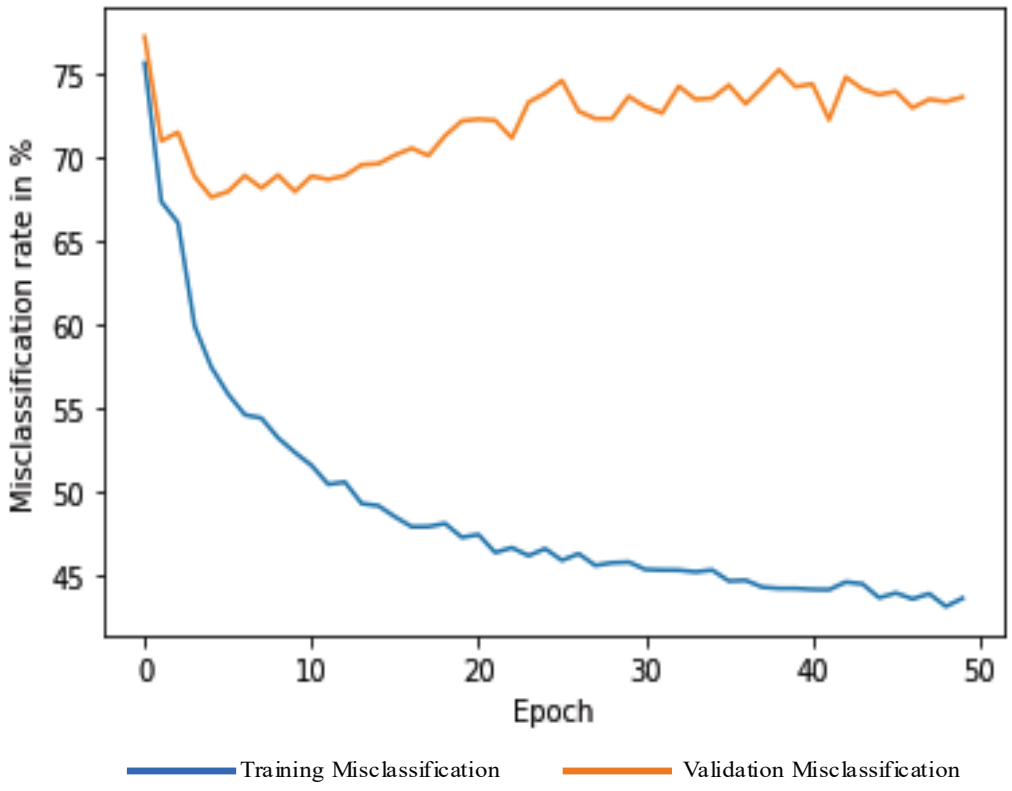


Figure 19: Learning curve for the best performing model on shifted sequences.

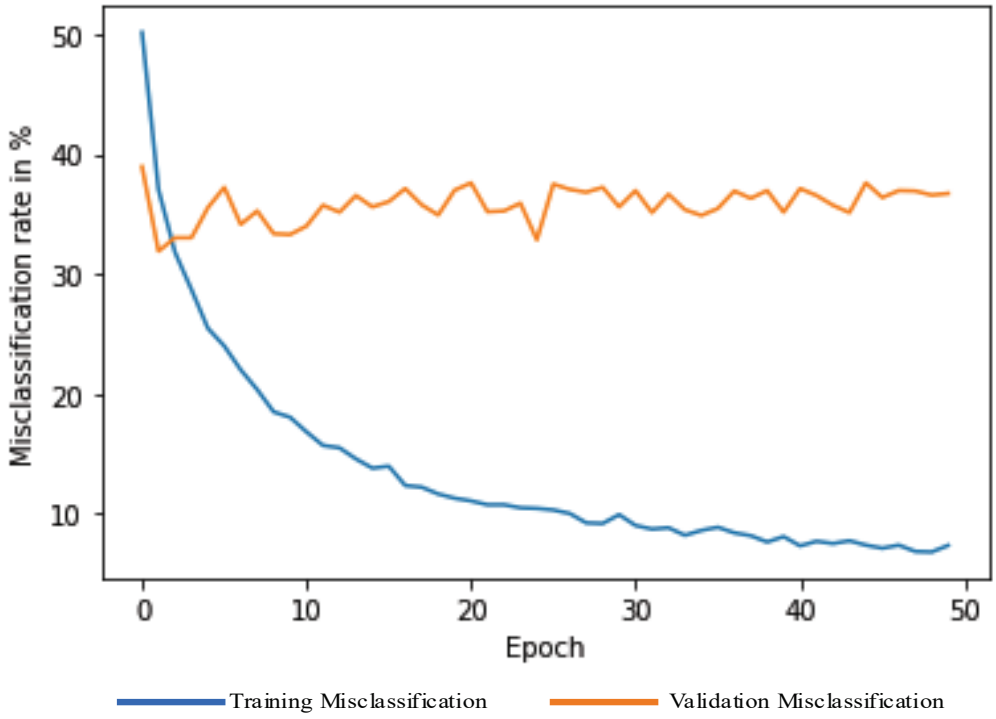
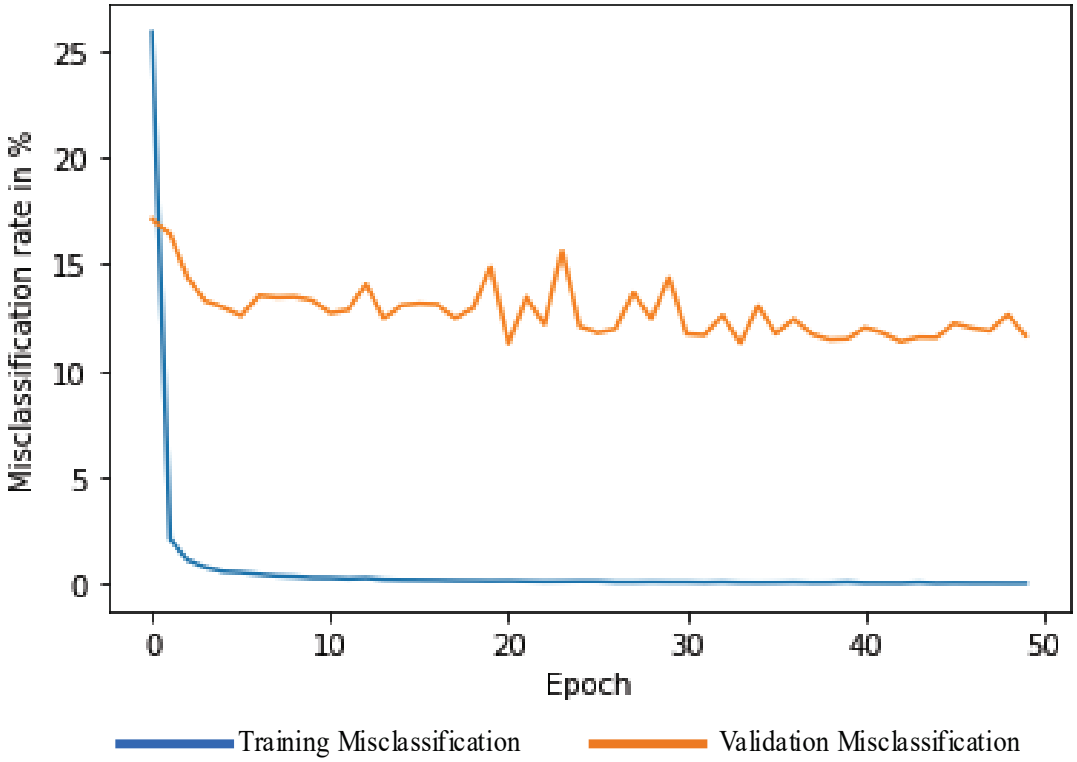


Figure 20: Learning curve for the best performing model on time ordered sequences.

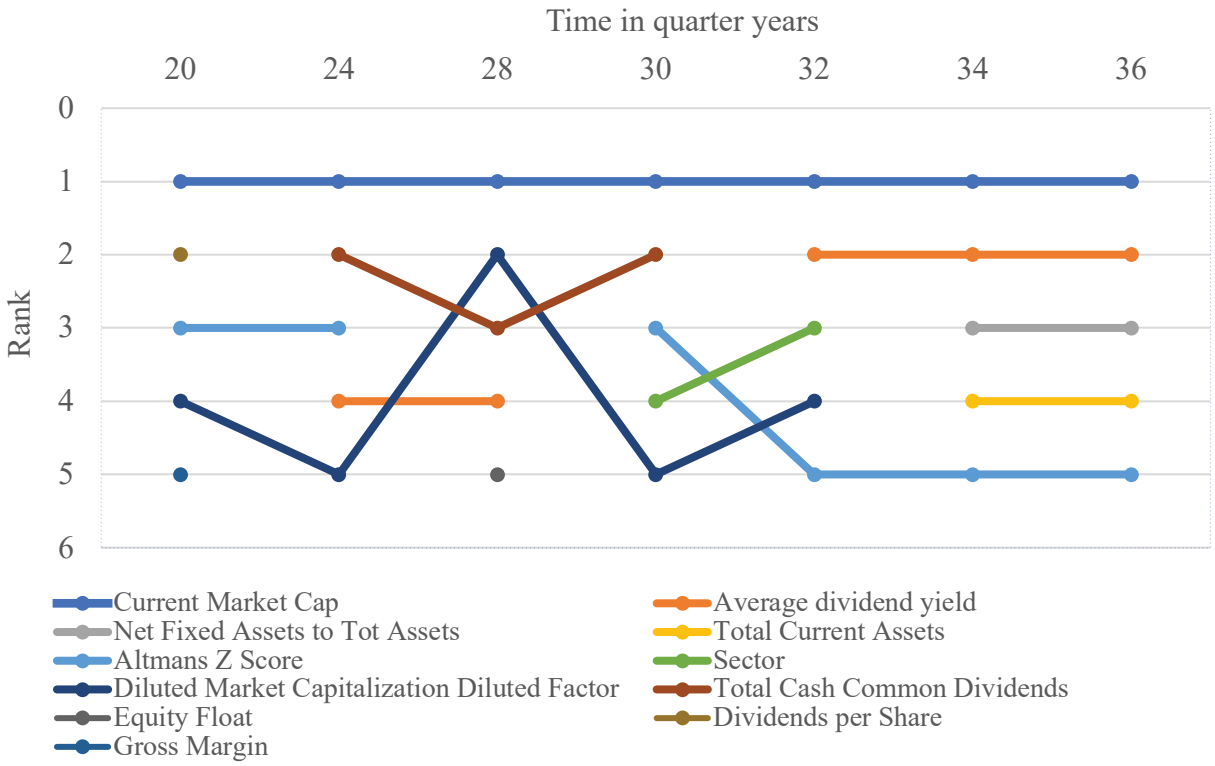


6.4 Variable importance observed over time

To preselect features when applying LSTM models, random forest was used to get important variables for each sequence length (see Table 7). Figure 21 is a bump chart showing the position of the top five variables from each sequence length and how their rank evolves over time. If a line is trending upwards, it implies that the feature is increasing in importance over time and vice versa. A discontinuity means that, the feature was no longer important enough to be in the top five in the next time period. A point means that, the feature was not in the top five most important features, but now has become important in that specific time period.

Figure 21 shows that the feature, Current Market Cap, remained the most important feature throughout the entire cumulative sequence. Diluted Market Capitalization Diluted Factor increased in importance in the earlier time periods, then dropped until it was no longer in the top five. Three features are only in the top five for one period and overall there are eleven distinct features observed in the top five features across the cumulative sequences.

Figure 21: Bump chart showing the rank of important variables over time in quarters (cumulative).



Chapter 7
Discussion of results

7.1 Discussion of Results

In this chapter a discussion of the results in relation to the objectives of the study is given together with their implications. The aim of this study was to compare the performance of random forests and LSTM neural networks in predicting corporate credit ratings in the USA. The need to do so arose from the fact that several studies have shown that random forests are better at predicting credit ratings than other models excluding LSTMs, while Golbayani, Wang and Florescu (2020) showed that LSTMs significantly outperform MLPs and CNNs. Unfortunately, their study did not include random forest.

7.2 Performance between random forest and LSTM

Based on the results in Chapter 6, the random forest models outperformed all the LSTM models by a wide margin. When the training sequence was varied but the prediction sequence was fixed at 2 quarters, the worst performing random forest model had a misclassification rate of 6.7% while the best performing LSTM model had a misclassification rate of 32.28%.

When the training sequence was equal to prediction sequence, the random forest models yet again outperformed the LSTM models by far (see Tables 11, 12, 13, 14 and 15). The worst performing random forest model had a misclassification rate of 20.6% while the best performing LSTM model had a misclassification rate of 39.47%. However, the misclassification rate between the best performing LSTM models did not vary as much as the misclassification rate between random forest models when the number of quarters being predicted were increased.

7.3 Impact of preselecting features

Figure 16 and Tables 13, 14 and 15 show that preselecting important features may be beneficial when using LSTMs. All the models except two of the best performing LSTM models had pre-selected features (see Tables 13,14 and 15). The test dataset results also show a slight improvement when input features are preselected. However, this is not comprehensive and

conclusive given that the difference in performance is only slight and the size of data on which it is based is small.

7.4 Impact of changing sequence format for LSTM models on performance

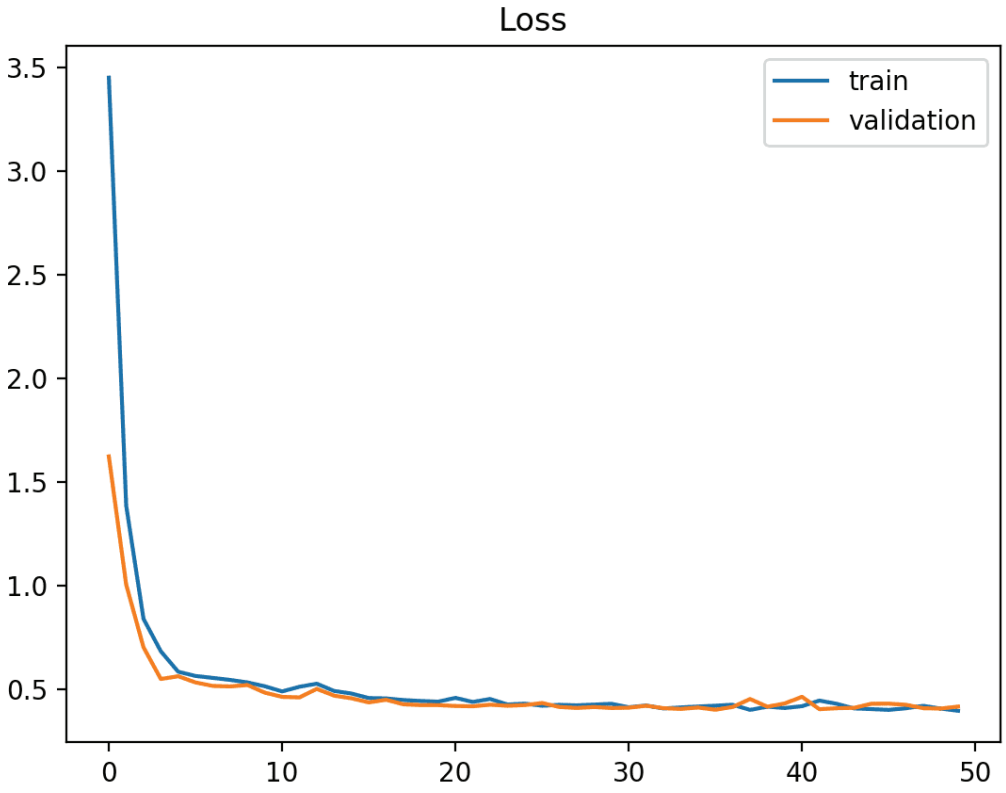
For the LSTM models, performance was compared when two different methods for feeding data to the model were used. The two approaches were "firm-ordered" data and "time-ordered" data. The results in Table 15 indicate that the performance for the time-ordered data was similar to the 'firm-ordered' data, suggesting that in this case, a larger sequence did not add any benefit to the performance. The best performing 'time-ordered' model had a misclassification rate of 28.35% when predicting credit ratings for 2 quarters ahead while using 2 quarters of training data. This was slightly better than the best performing model when 'firm-ordered' data was used in predicting the same sequence despite the fact that, the time-ordered data had a sequence length of 254 while the corresponding firm-ordered data had a sequence length of 2. The downside to using time ordered data is that although the sequence length is longer, the number of samples being fed in each training iteration is were fewer than when the sequences is in firm order.

7.5 Analysis of the models' learning curves

Although the results show that random forests outperformed LSTM models, LSTM models are a lot complex and difficult to tune. In this study, 16,473 LSTM models were run in an attempt to identify the best performing model. It is also important to try to understand why these models may not have scored as accurately as expected and if at all they could potentially be improved in future studies or if this information can be applied in another context.

A simple way to understand what is happening with the performance of a model is to plot learning curves. Learning curves show the training performance compared with the cross validation performance over time (epochs). These graphs can help identify if a model is a good fit to the data. An example of a good fit generally has the validation metric tracking closely to the training metric and the performance remains fairly similar across the epochs (See Figure 22).

Figure 22: An example of a good fit learning curve



Source: (Brownlee, 2018)

The graphs for the best performing models are shown in section 6.3.6. In general, they all show the training misclassification rate decreasing rapidly until the training misclassification rate is less than the validation misclassification rate. Beyond this point the training misclassification rate improves at a slower rate. The validation misclassification rate stops improving after the first few epochs and fluctuates around the same level. This is indicative of a an unrepresentative validation set. It is frequently seen when more data is needed to validate the performance of the model and sometimes when the validation set is easier than the validation dataset (Brownlee, 2018). In this case the training metric is better than the validation metric, thus an easier validation set can be ruled out, concluding that more data is needed to properly validate the models.

7.6 Data quality and availability issues

This is a problem that is rather difficult to solve since in order to get more data either more firms need to be added or more time periods. Adding more time periods means that the focus on validation is not fully on the latest data. Since financial markets are time sensitive there is more benefit in being able to predict the latest data. On the other hand, adding firms may be impossible because there is a limited number of firms that are rated, of these firms, some large firms have not experienced any credit rating changes and thus would not be suitable candidates in training a model.

Additionally, not all credit rating and financial data is easily available, some firms' credit rating data is difficult to access and often requires expensive memberships. It is compounded by the fact that market conditions are changing as technology disrupts the financial sector, thus rendering historic data to be less useful. This study alludes to this fact in several places, firstly it shows that short sequences even of two quarters can be used to predict credit ratings reasonably (see Tables 11, 12, 13, 14 and 15). It also shows that important features change over time and fluctuate even when the features are identified cumulatively (see Figure 21).

Additionally, it could also feed into the existing argument that credit ratings can behave like self-fulfilling prophecies and that the credit ratings are not representative of the conditions captured by the markets. The lack of data is something that cannot be ruled out easily. However this study proposes a method to overcome the notion of lack of data. This is discussed in the future work section in Chapter 8.

7.7 Summary and practical applications

The results from this study show that random forest models outperform LSTM models based when predicting corporate credit ratings in the USA. This study also showed that there is in fact an optimal number of timesteps when predicting corporate credit ratings. Random forest based models performed best when 6 quarters of input data was used to predict 2 quarters ahead, while LSTM models performed best when 4 quarters of input data was used to predict 2 quarters ahead. This finding also implies recent data is more important at predicting credit ratings than historic data. Finally, this study found that there might be some benefit to preselecting features when fitting LSTM models as all the best LSTM models contained preselected features. However, it does not negate the ability of LSTMs to identify important

features as the performance difference on average between models with preselected features and models without any feature selection was low.

The findings in this study are also comparable to the existing literature. The RFs in this study had a lower misclassification rate than related studies. However, the LSTMs in this study did not perform as well as the LSTMs in related studies. The LSTMs in this study did however, perform comparably to the CNNs and the MLPs on Table 3. This performance penalty may be explained by the fact that the LSTM models in this study are able to predict across multiple sectors (10) using only 2 or 4 quarters of input data. Additionally, the LSTMs in this study were able to predict with reasonable accuracy two and a half years into the future (10 quarters) using less training data than all the studies it is being compared to.

A notable difference between the models in this study compared to related studies was the number of predicted classes. Whereas Golbayani, Wang and Florescu, (2020) used the distinct number of credit ratings in their data as the number of classes being predicted and focused on individual sectors, this approach would not generalize well to new information for two reasons. Firstly, new information may contain credit ratings in a class that was unseen in their models and thus would not be predicted correctly. Additionally, the models built would be too specific to their sector and might only be practically useful if the user was interested in that specific sector and nothing else.

On the contrary, the models in this study are highly generalisable and would be much more useful in a practical scenario. This is because they can predict across ten sectors and across the entire range of credit ratings. For example, a RF model that uses a small amount of recent data that can be run on a home user's laptop that is able to predict credit ratings six months into the future with 95% accuracy across ten sectors could be highly leveraged by a business or an investor. Such a model could be run daily as soon as new financial data is available.

Additionally, a similar model with similar computational cost that can predict with some accuracy, in this case about 52%, up to two and a half years into the future can be used to paint a broad picture of what the outlook might be. The predictions from the more accurate short term models can be supplemented with a broad long term estimate to further guide the investor/user on what strategies to apply.

These decisions would ultimately lie with the user and what their preferred outcome would be. If the user has access to a lot of data or can access data cheaply, and may be managing large

portfolios where minor deviations could result in large financial losses, they may be better off using multiple models all at once in an ensemble. If however, the user is a small business or a private trader with access to limited data they could achieve similar results by using only the short term models.

Chapter 8

Conclusion and future research

This study compared the performance between LSTM neural networks and random forests in predicting S&P's credit ratings for companies in the USA. The basis of this study arose from a gap in the literature in comparing these two algorithms, as discussed in the literature review, several studies show that random forests outperform MLPs in predicting credit ratings and recent study has shown that LSTM significantly outperform MLPs and CNNs.

This study used data obtained from the Bloomberg terminal, containing financial data for 127 US firms rated by S&P with 123 explanatory variables over 40 quarters during the period of 2010 to 2019. This study used quarterly data instead of yearly data to examine whether accurate predictions can be made using less but more recent data rather than a large dataset that contains largely historic data, additionally, using quarterly data allows the model to predict a rating change closer to when it actually happens rather than having up to a year's delay in observing the change.

In addition to using quarterly data, this study took guidance from recent studies suggesting that a complete dataset without missing variables and zero imputations might allow neural networks to perform better. Thus, this study implements the multiple imputation by chained equations method along with exploratory data analysis and theoretical guidance from the literature to create a complete dataset for 10 of the 11 GICS sectors in the S&P excluding the financials sector. This study, then performed feature selection on the entire dataset to identify the top variables in this dataset, these variables were then used to compare the effectiveness of the LSTM algorithm in identifying important variables.

This study then applied the random forest model in a temporal split of the data, varying the training quarters while holding the testing quarters fixed at 2, it identified the best performing training set was using 6 quarters.

A similar experiment was conducted with the LSTM models and this study found the 4 quarter sequences to be the best performing. It also performed experiment where the number of testing quarters were kept the same as the number of training quarters and for both RF and LSTMs the best performing sequence length was 4 quarters. Furthermore, the best performing LSTM

models arose from having preselected features, however, this does not conclusively prove that LSTMs are not able to identify important variables, rather it implies, it was unable to do so in this case and having a feature selection method prior to training may improve performance and reduce computational cost.

Finally, this study showed that random forest models were better at predicting credit ratings than LSTM neural networks, with the best performing random forest model having a misclassification rate of 3.54% and the best performing LSTM model a misclassification rate of 28.35%. These results were obtained from a holdout set that contained four quarters, the first two quarters for training and the last two quarters for testing.

This study contributes to the literature in several ways, the first is the gap of comparison between what is shown to be the best performing neural network in predicting credit ratings compared with the random forest algorithm which has been shown to outperform standard multilayer perceptrons in the literature. Secondly, it uses a recent dataset with ten GICS sectors instead of focusing only on one sector. Thirdly, it presents a complete dataset with quarterly data and with all the missing data points imputed with actual values using the MICE algorithm instead of zero imputation or removal of all missing points. Finally, this study showed that when using quarterly data, smaller datasets can be used to predict farther in the future and closer to the actual rating change, if they contain the most recent data.

The discussion points in this study show that in application all the models trained and tested in this study would add tremendous value to a user in a practical situation, this is because they are lightweight and require little data. However, the LSTMs did not perform as well as the random forests and the LSTM models in the study performed by Golbayani, Wang and Florescu, (2020). This performance gap was identified as being due to a misrepresentative dataset. LSTMs are complex algorithms and could benefit from having more data.

This study therefore proposes several suggestions for future research. The first is to use variational autoencoders to generate simulated data so that a model can still be trained with the most recent two quarters of data but there are more examples for the data to learn from. Secondly, this study proposes the use of transformers and to compare them with random forests. Thirdly, it proposes the use of an ensemble of models. One kind of ensemble could be created by using models that can predict in the long run with data containing yearly data in conjunction with models predicting on quarterly data. Another kind of ensemble could focus

on wider bands of credit ratings first, for instance, the first model could separate rating based on investment or non-investment grade, followed by more granular models for each rating class. Finally, it suggests using models that can predict credit ratings based on a combination of sentiment analysis, financial data and socioeconomic data.

Bibliography

1. Addo, P.M., Guegan, D. and Hassani, B., 2018. Credit risk analysis using machine and deep learning models. *Risks*, *6*(2), p.38.
2. Agbehadji, I.E., Millham, R., Fong, S.J. and Yang, H., 2018, September. Kestrel-Based Search Algorithm (KSA) for Parameter Tuning Unto Long Short Term Memory (LSTM) Network for Feature Selection in Classification of High-Dimensional Bioinformatics Datasets. In *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 15-20). IEEE.
3. Alfaro, E., García, N., Gámez, M. and Elizondo, D., 2008. Bankruptcy forecasting: An empirical comparison of AdaBoost and neural networks. *Decision Support Systems*, *45*(1), pp.110-122.
4. Althelaya, K.A., El-Alfy, E.S.M. and Mohammed, S., 2018, April. Evaluation of bidirectional lstm for short-and long-term stock market prediction. In *2018 9th international conference on information and communication systems (ICICS)* (pp. 151-156). IEEE.
5. Altman, E., and Katz, S. 1976. Statistical bond rating classification using financial and accounting data. In Michael Schiff and George Sorter (eds.), Proceedings of the Conference on Topical
6. Ambler, G., Omar, R.Z. and Royston, P., 2007. A comparison of imputation techniques for handling missing predictor values in a risk model with a binary outcome. *Statistical methods in medical research*, *16*(3), pp.277-298.
7. Ammer, J. and Clinton, N., 2004. Good news is no news? The impact of credit rating changes on the pricing of asset-backed securities. *The Impact of Credit Rating Changes on the Pricing of Asset-Backed Securities (July 2004). FRB International Finance Discussion Paper*, (809).
8. Arouri, C., Nguifo, E.M., Aridhi, S., Roucelle, C., Bonnet-Loosli, G. and Tsopzé, N., 2014. Towards a constructive multilayer perceptron for regression task using non-parametric clustering. A case study of Photo-Z redshift reconstruction. *arXiv preprint arXiv:1412.5513*.
9. Azur, M.J., Stuart, E.A., Frangakis, C. and Leaf, P.J., 2011. Multiple imputation by chained equations: what is it and how does it work?. *International journal of methods in psychiatric research*, *20*(1), pp.40-49.

10. Baneshi, M.R. and Talei, A.R., 2012. Does the missing data imputation method affect the composition and performance of prognostic models?. *Iranian Red Crescent Medical Journal*, *14*(1), p.31.
11. Banfield, R.E., Hall, L.O., Bowyer, K.W. and Kegelmeyer, W.P., 2006. A comparison of decision tree ensemble creation techniques. *IEEE transactions on pattern analysis and machine intelligence*, *29*(1), pp.173-180.
12. Bengio, Y., Simard, P. and Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), pp.157-166.
13. Bennett, D.A., 2001. How can I deal with missing data in my study?. *Australian and New Zealand journal of public health*, *25*(5), pp.464-469.
14. Bhojraj, S. and Sengupta, P., 2003. Effect of corporate governance on bond ratings and yields: The role of institutional investors and outside directors. *The journal of Business*, *76*(3), pp.455-475.
15. Borensztein, E., Cowan, K. and Valenzuela, P., 2013. Sovereign ceilings "lite"? The impact of sovereign ratings on corporate ratings. *Journal of Banking & Finance*, *37*(11), pp.4014-4024.
16. Breiman, L., 1996. Bagging predictors. Machine learning, 24(2), pp.123-140.
17. Breiman, L., 2001. Random forests. Machine learning, 45(1), pp.5-32.
18. Brennan, D. and Brabazon, A., 2004. Corporate Bond Rating Using Neural Networks. In *IC-AI* (pp. 161-167).
19. Brooks, R., Faff, R.W., Hillier, D. and Hillier, J., 2004. The national market impact of sovereign rating changes. *Journal of banking & finance*, *28*(1), pp.233-250.
20. Brownlee, J., 2018. Better deep learning: train faster, reduce overfitting, and make better predictions. Machine Learning Mastery.
21. Buuren, S.V. and Groothuis-Oudshoorn, K., 2010. mice: Multivariate imputation by chained equations in R. *Journal of statistical software*, pp.1-68.
22. Calin, O., 2020. Recurrent Neural Networks. In Deep Learning Architectures (pp. 543-559). Springer, Cham.
23. Carlenius, B., Døvik, E.K., Kolberg, J.K., Waage, K. and Aanes, B., 2017. Corporate Credit Rating using Deep Learning with Genetic Algorithms.
24. Chandra, D.K., Ravi, V. and Bose, I., 2009. Failure prediction of dotcom companies using hybrid intelligent techniques. Expert Systems with Applications, 36(3), pp.4830-4837.

25. Chen, K., Zhou, Y. and Dai, F., 2015, October. A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE international conference on big data (big data)* (pp. 2823-2824). IEEE.
26. Chen, Y.S., 2012. Classifying credit ratings for Asian banks using integrating feature selection and the CPDA-based rough sets approach. *Knowledge-Based Systems*, *26*, pp.259-270.
27. Chhabra, G., Vashisht, V. and Ranjan, J., 2017. A comparison of multiple imputation methods for data with missing values. *Indian Journal of Science and Technology*, *10*(19), pp.1-7.
28. Chow, J.C., 2018. Analysis of financial credit risk using machine learning. *arXiv preprint arXiv:1802.05326*.
29. Corrêa, D.C., Salvadeo, D., Levada, A., Saito, J.H., Mascarenhas, N. and Moreira, J., 2008. Using lstm network in face classification problems.
30. Delahunty, A., 2004. *Artificial immune systems for the prediction of corporate failure and classification of corporate bond ratings* (Doctoral dissertation, University College Dublin, Graduate School of Business).
31. Dietterich, T.G., 1998. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine learning*, *32*, pp.1-22.
32. Dong, Y. and Peng, C.Y.J., 2013. Principled missing data methods for researchers. *SpringerPlus*, *2*(1), pp.1-17.
33. Driss, S.B., Soua, M., Kachouri, R. and Akil, M., 2017, May. A comparison study between MLP and Convolutional Neural Network models for character recognition. In *Real-Time Image and Video Processing 2017* (Vol. 10223, p. 1022306). International Society for Optics and Photonics.
34. Ederington, L.H., 1985. Classification models and bond ratings. Financial review, 20(4), pp.237-262.
35. Frydman, H., Altman, E.I. and Kao, D.L., 1985. Introducing recursive partitioning for financial classification: the case of financial distress. The Journal of Finance, 40(1), pp.269-291.
36. Fu, K., Cheng, D., Tu, Y. and Zhang, L., 2016, October. Credit card fraud detection using convolutional neural networks. In *International Conference on Neural Information Processing* (pp. 483-490). Springer, Cham.

37. Galil, K., 2003, October. The quality of corporate credit rating: an empirical investigation. In *EFMA 2003 Helsinki Meetings*.
38. Garavaglia, S., 1991, January. An application of a counter-propagation neural network: simulating the Standard and Poor's Corporate Bond Rating system. In *Proceedings First International Conference on Artificial Intelligence Applications on Wall Street* (pp. 278-279). IEEE Computer Society.
39. Gareth, J., Daniela, W., Trevor, H. and Robert, T., 2013. An introduction to statistical learning: with applications in R. Spinger.
40. Geenens, G., 2011. Curse of dimensionality and related issues in nonparametric functional regression. *Statistics Surveys*, *5*, pp.30-43.
41. Gelman, A. and Hill, J., 2006. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.
42. Gers, F.A., Schmidhuber, J. and Cummins, F., 2000. Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), pp.2451-2471.
43. Giesecke, K. and Weber, S., 2004. Cyclical correlations, credit contagion, and portfolio losses. *Journal of Banking & Finance*, *28*(12), pp.3009-3036.
44. Golbayani, P., Florescu, I. and Chatterjee, R., 2020. A comparative study of forecasting corporate credit ratings using neural networks, support vector machines, and decision trees. *The North American Journal of Economics and Finance*, *54*, p.101251.
45. Golbayani, P., Wang, D. and Florescu, I., 2020. Application of deep neural networks to assess corporate credit rating. *arXiv preprint arXiv:2003.02334*.
46. Graham, J.W., Olchowski, A.E. and Gilreath, T.D., 2007. How many imputations are really needed? Some practical clarifications of multiple imputation theory. *Prevention science*, *8*(3), pp.206-213.
47. Ha, V.S. and Nguyen, H.N., 2016. Credit scoring with a feature selection approach based deep learning. In *MATEC Web of Conferences* (Vol. 54, p. 05004). EDP Sciences.
48. Hajek, P. and Michalak, K., 2013. Feature selection in corporate credit rating prediction. Knowledge-Based Systems, 51, pp.72-84.
49. Hájek, P. and Olej, V., 2014, September. Predicting firms' credit ratings using ensembles of artificial immune systems and machine learning–an over-sampling approach. In *IFIP International Conference on Artificial Intelligence Applications and Innovations* (pp. 29-38). Springer, Berlin, Heidelberg.

50. Hájek, P., 2012. Credit rating analysis using adaptive fuzzy rule-based systems: an industry-specific approach. *Central European Journal of Operations Research*, *20*(3), pp.421-434.
51. Hastie, T., Tibshirani, R., Friedman, J.H. and Friedman, J.H., 2009. The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.
52. Heidt, K., 2019. Comparison of Imputation Methods for Mixed Data Missing at Random.
53. Ho, T.K., 1995, August. Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition (Vol. 1, pp. 278-282). IEEE.
54. Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.
55. Horrigan, J.O., 1966. The determination of long-term credit standing with financial ratios. *Journal of Accounting Research*, pp.44-62.
56. Huang, S.C., 2011. Using Gaussian process based kernel classifiers for credit rating forecasting. *Expert Systems with Applications*, *38*(7), pp.8607-8611.
57. Huang, Y.L. and Shen, C.H., 2015. The sovereign effect on bank credit ratings. *Journal of Financial Services Research*, *47*(3), pp.341-379.
58. Huang, Y.M., Hung, C.M. and Jiau, H.C., 2006. Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem. *Nonlinear Analysis: Real World Applications*, *7*(4), pp.720-747.
59. Huang, Z., Chen, H., Hsu, C.J., Chen, W.H. and Wu, S., 2004. Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision support systems*, *37*(4), pp.543-558.
60. Hung, C.H.D., Banerjee, A. and Meng, Q., 2017. Corporate financing and anticipated credit rating changes. *Review of quantitative finance and accounting*, *48*(4), pp.893-915.
61. Hwang, R.C., Chung, H. and Chu, C.K., 2010. Predicting issuer credit ratings using a semiparametric method. *Journal of Empirical Finance*, *17*(1), pp.120-137.
62. Kaplan, R.S. and Urwitz, G., 1979. Statistical models of bond ratings: A methodological inquiry. *Journal of business*, pp.231-261.
63. Kim, K.S., 2005. Predicting bond ratings using publicly available information. *Expert Systems with Applications*, *29*(1), pp.75-81.
64. Kim, Y. and Nabar, S., 2003. Why do stock prices react to bond rating downgrades?. *Managerial Finance*.

65. King, Peter, and Heath Tarbert. "Basel III: an overview." *Banking & Financial Services Policy Report* 30, no. 5 (2011): 1-18.
66. Kirkegaard, E.O.W, 2016, 'R functions for analyzing missing data', *Clear Language, Clear Mind,* 14 April, Available at: https://emilkirkegaard.dk/en/2016/04/r-functions-for-analyzing-missing-data/(Accesed: 29 May 2021)
67. Koutanaei, F.N., Sajedi, H. and Khanbabaei, M., 2015. A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring. *Journal of Retailing and Consumer Services*, *27*, pp.11-23.
68. Kumar, K. and Bhattacharya, S., 2006. Artificial neural network vs linear discriminant analysis in credit ratings forecast. *Review of Accounting and Finance*.
69. Kuo, F.Y. and Sloan, I.H., 2005. Lifting the curse of dimensionality. *Notices of the AMS*, *52*(11), pp.1320-1328.
70. Lawrence Fisher, "Determinants of Risk Premiums on Corporate Bonds," Journal of Political Economy, June 1959, pp. 217-237.
71. Lee, J., Bahri, Y., Novak, R., Schoenholz, S.S., Pennington, J. and Sohl-Dickstein, J., 2017. Deep neural networks as gaussian processes. arXiv preprint arXiv:1711.00165.
72. Lehmann, B., 2003. Is it worth the while? The relevance of qualitative information in credit rating. The Relevance of Qualitative Information in Credit Rating (April 17, 2003). EFMA.
73. Lin, W.Y., Hu, Y.H. and Tsai, C.F., 2011. Machine learning in financial crisis prediction: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(4), pp.421-436.
74. Liu, Y. and Brown, S.D., 2013. Comparison of five iterative imputation methods for multivariate classification. Chemometrics and Intelligent Laboratory Systems, 120, pp.106-115.
75. Livingstone, D.J., Manallack, D.T. and Tetko, I.V., 1997. Data modelling with neural networks: advantages and limitations. *Journal of computer-aided molecular design*, *11*(2), pp.135-142.
76. Madeh Piryonesi, S. and El-Diraby, T.E., 2021. Using Machine Learning to Examine Impact of Type of Performance Indicator on Flexible Pavement Deterioration Modeling. Journal of Infrastructure Systems, 27(2), p.04021005.
77. Matthies, A.B., 2013. *Empirical research on corporate credit-ratings: a literature review* (No. 2013-003). SFB 649 Discussion paper.

78. McCarthy, J.E. and Melicher, R.W., 1988. Analysis of bond rating changes in a portfolio context. *Quarterly Journal of Business and Economics*, pp.69-86.
79. McKelvey, R., and Zavoina, W. 1975. A statistical model for the analysis of ordinal level dependent variables. Journal of Mathematical Sociology 4 (Summer): 103-20
80. McKnight, P.E., McKnight, K.M., Sidani, S. and Figueredo, A.J., 2007. *Missing data: A gentle introduction*. Guilford Press.
81. McNeish, D., 2017. Missing data methods for arbitrary missingness with small samples. *Journal of Applied Statistics*, *44*(1), pp.24-39.
82. Medina, E., Petraglia, M.R., Gomes, J.G.R. and Petraglia, A., 2017. Comparison of CNN and MLP classifiers for algae detection in underwater pipelines. In *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)* (pp. 1-6). IEEE.
83. Morey, M.R., 2002. Mutual fund age and Morningstar ratings. *Financial Analysts Journal*, *58*(2), pp.56-63.
84. Nazmi, N. and Ramirez, M.D., 1997. Public and private investment and economic growth in Mexico. *Contemporary Economic Policy*, *15*(1), pp.65-75.
85. Nelson, D.M., Pereira, A.C. and de Oliveira, R.A., 2017, May. Stock market's price movement prediction with LSTM neural networks. In *2017 International joint conference on neural networks (IJCNN)* (pp. 1419-1426). IEEE.
86. Novotná, M., 2012, October. The use of different approaches for credit rating prediction and their comparison. In *Proceedings of the 6th International Conference on Managing and Modelling of Financial Risks* (pp. 448-457).
87. Olah, C., 2015. Understanding lstm networks. Available at: https://colah.github.io/posts/2015-08-Understanding-LSTMs/index.html (Accessed: 30th August 2022)
88. Ozturk, H., Namli, E. and Erdal, H.I., 2016. Reducing overreliance on sovereign credit ratings: which model serves better?. *Computational Economics*, *48*(1), pp.59-81.
89. Pascanu, R., Mikolov, T. and Bengio, Y., 2013, May. On the difficulty of training recurrent neural networks. In International conference on machine learning (pp. 1310-1318). PMLR.
90. Peng, C.J., Harwell, M., Liou, S.M., Ehman, L.H. and Sawilowsky, S., 2006. Real data analysis. *Advances in missing data methods and implications for educational research*, pp.31-78.
91. Pettit, J., 2004. The new world of credit ratings. *Available at SSRN 593522*.

92. Pinches, G., and Mingo, K. 1975. A note on the role of subordination in determining industrial bond ratings. Journal of Finance 30 (March): 201-6.
93. Pinches, G.E. and Mingo, K.A., 1973. A multivariate analysis of industrial bond ratings. *The journal of Finance*, *28*(1), pp.1-18.
94. Pogue, T.F. and Soldofsky, R.M., 1969. What's in a Bond Rating. *Journal of financial and quantitative analysis*, *4*(2), pp.201-228.
95. Raeesy, Z., Gillespie, K., Ma, C., Drugman, T., Gu, J., Maas, R., Rastrow, A. and Hoffmeister, B., 2018, December. Lstm-based whisper detection. In *2018 IEEE Spoken Language Technology Workshop (SLT)* (pp. 139-144). IEEE.
96. Raghunathan, T.E., Solenberger, P.W. and Van Hoewyk, J., 2002. IVEware: Imputation and variance estimation software. *Ann Arbor, MI: Survey Methodology Program, Survey Research Center, Institute for Social Research, University of Michigan*.
97. Reed, R. and MarksII, R.J., 1999. Neural smithing: supervised learning in feedforward artificial neural networks. MIT Press
98. Rosenberg, E. and Gleit, A., 1994. Quantitative methods in credit management: a survey. Operations research, 42(4), pp.589-613.
99. Royston, P. and White, I.R., 2011. Multiple imputation by chained equations (MICE): implementation in Stata. *J Stat Softw*, *45*(4), pp.1-20.
100. Rubin, D.B., 1976. Inference and missing data. Biometrika, 63(3), pp.581-592.
101. Ryser, M. and Denzler, S., 2009. Selecting credit rating models: a cross-validation-based comparison of discriminatory power. *Financial Markets and Portfolio Management*, *23*(2), pp.187-203.
102. S&P Dow Jones Indices, 2020. Equity S&P 500. Available at: https://www.spglobal.com/spdji/en/indices/equity/sp-500/#data (Accessed: 30th June 2020)
103. S&P Global Ratings, 2021a. S&P Global Ratings Definitions. Available at: https://disclosure.spglobal.com/ratings/en/regulatory/article/-/view/sourceId/504352 (Accessed: 30th April 2022)
104. S&P Global Ratings, 2021b. Default, Transition, and Recovery: 2020 Annual Global Corporate Default And Rating Transition Study. Available at: https://www.spglobal.com/ratings/en/research/articles/210407-default-transition-and-recovery-2020-annual-global-corporate-default-and-rating-transition-study-11900573 (Accessed: 30th April 2022)

105. Saitoh, F., 2016, December. Predictive modeling of corporate credit ratings using a semi-supervised random forest regression. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 429-433). IEEE.
106. Satchidananda, S.S. and Simha, J.B., 2006. Comparing decision trees with logistic regression for credit risk analysis. *International Institute of Information Technology, Bangalore, India*.
107. Schafer, J.L., 1999. Multiple imputation: a primer. *Statistical methods in medical research*, *8*(1), pp.3-15.
108. Seddik, A., 2015. *Corporate Bond Valuation and Credit Spreads: Lessons from the Financial Crisis* (Doctoral dissertation).
109. Shin, K.S. and Han, I., 2001. A case-based approach using inductive indexing for corporate bond rating. *Decision Support Systems*, *32*(1), pp.41-52.
110. Steiner, M. and Heinke, V.G., 2001. Event study concerning international bond price effects of credit rating actions. *International Journal of Finance & Economics*, *6*(2), pp.139-157.
111. Sun, S., Wei, Y. and Wang, S., 2018, June. Adaboost-lstm ensemble learning for financial time series forecasting. In *International Conference on Computational Science* (pp. 590-597). Springer, Cham.
112. Sylla, R., 2002. An historical primer on the business of credit rating. In *Ratings, rating agencies and the global financial system* (pp. 19-40). Springer, Boston, MA.
113. Tabachnick, B.G., Fidell, L.S. and Ullman, J.B., 2007. *Using multivariate statistics* (Vol. 5, pp. 481-498). Boston, MA: Pearson.
114. Tsai, C.F. and Chen, M.L., 2010. Credit rating by hybrid machine learning techniques. *Applied soft computing*, *10*(2), pp.374-380.
115. U.S. Securities and exchange commission, 2022. Exchange Act Reporting and Registration. Available at: https://www.sec.gov/education/smallbusiness/goingpublic/exchangeactreporting (Accessed: 30th April 2022)
116. Van Buuren, S., 2018. *Flexible imputation of missing data*. CRC press.
117. Wallis, M., Kumar, K. and Gepp, A., 2019. Credit Rating Forecasting Using Machine Learning Techniques. In Managerial Perspectives on Intelligent Big Data Analytics (pp. 180-198). IGI Global.
118. Wang, G., Ma, J., Huang, L. and Xu, K., 2012. Two credit scoring models based on dual strategy ensemble trees. *Knowledge-Based Systems*, *26*, pp.61-68.

119. West, D., 2000. Neural network credit scoring models. *Computers & Operations Research*, *27*(11-12), pp.1131-1152.
120. West, R.R., 1970. An alternative approach to predicting corporate bond ratings. *Journal of Accounting Research*, pp.118-125.
121. White, I.R., Royston, P. and Wood, A.M., 2011. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, *30*(4), pp.377-399.
122. White, L.J., 2010. Markets: The credit rating agencies. *Journal of Economic Perspectives*, *24*(2), pp.211-26.
123. Wilamowski, B.M., 2009. How Not to Be Frustrated with Neural Networks. *eng. auburn. edu, no. December*, pp.56-63.
124. Yang, J. and Zhang, L., 2011. The Impacts of Sovereign Credit Ratings on Exchange Rates-Evidence from Eurozone Sovereign Debt Crisis.
125. Ye, Y., Liu, S. and Li, J., 2008, May. A multiclass machine learning approach to credit rating prediction. In *2008 International Symposiums on Information Processing* (pp. 57-61). IEEE.
126. Yeh, C.C., Lin, F. and Hsu, C.Y., 2012. A hybrid KMV model, random forests and rough set theory approach for credit rating. *Knowledge-Based Systems*, *33*, pp.166-172.
127. Yi J, Lee J, Kim KJ, Hwang SJ, Yang E. Why not to use zero imputation? correcting sparsity bias in training neural networks. arXiv preprint arXiv:1906.00150. 2019 Jun 1.
128. Yu, Z., Ramanarayanan, V., Suendermann-Oeft, D., Wang, X., Zechner, K., Chen, L., Tao, J., Ivanou, A. and Qian, Y., 2015, December. Using bidirectional LSTM recurrent neural networks to learn high-level abstractions of sequential features for automated scoring of non-native spontaneous speech. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)* (pp. 338-345). IEEE.
129. Zhou, Z.H., 2009. Ensemble Learning. *Encyclopedia of biometrics*, *1*, pp.270-273.
130. Zhou, Z.H., 2012. Ensemble methods: foundations and algorithms. CRC press.

Appendix A

Table 1: Table listing the input variables defined in Hajek and Michalak's (2013) study on credit ratings.

Leverage ratios	**Activity ratios**	**Size of company**
Book value to equity	Enterprise value to sales	Total assets
Book debt to total capital	Growth in non cash working capital	Total capital
Enterprise value to total capital	Enterprise value to trailing sales	Sales (last year)
Enterprise value to book value	Sales to net worth	12-Moth trailing sales
Market capitalization to total debt	Sales to total assets	Cash flow
Total debt	Operating revenue to total assets	Equity
Cash flow to total debt	Working capital to sales	Enterprise value
Market debt to equity	Cash to sales	Firm value
Market debt to total capital	Non-cash working capital to sales	Capital expenditures
Net gearing		Size class
	Asset structure	Market capitalization
Market value ratios	Fixed assets to total assets	Trading volume
3-Year stock price variation	Intangible assets to total assets	No. of shares outstanding
Beta regression coefficient (3 year)	Working capital to total assets	
Value line beta	Depreciation	**Profitability ratios**
The correlation of stock returns with market index		Earnings before interest and taxes (EBIT)
Dividends	**Business situation**	Earnings after taxes
Dividends to stock price	Effective tax rate	Net income (NI)
Earnings per share (EPS)	Growth in sales last year	12-Month trailing NI
Growth in earnings per share (last 5 years)	Expected. growth in Sales (5 years)	Net margin
Expected growth in EPS (next 5 years)	SG&A expenditures	Operating margin
Stock price to cash flow		Return on total assets
Stock price to earnings	**Liquidity ratios**	Return on equity
12-Month trailing stock price to earnings	Current ratio	Return on capital
Forward stock price to earnings	Cash ratio	EBIT increased by depreciation and amortization (EBITDA)
Stock price to earnings to EPS growth	Cash to firm value	Enterprise value to EBITDA
Price to book value ratio	Cash	High/low stock price
Retained earnings	Non-cash working capital	Enterprise value to EBIT
Reinvestment rate		Retained earnings to total assets
Payout ratio	**Corporate reputation**	
Stock price to sales	Shares held by mutual funds	
Stock price	Shares held by insiders	

Table 2: Confusion Matrix - Model MIN22.

	1	2	3	4	Error
1	53	1	0	0	1.85%
2	3	95	0	0	3.06%
3	0	7	64	0	9.86%
4	1	0	0	30	3.23%
Total	57	103	64	30	4.72%

Table 3: Confusion Matrix - Model AVG22.

	1	2	3	4	Error
1	53	1	0	0	1.85%
2	3	95	0	0	3.06%
3	1	6	63	1	11.27%
4	0	1	1	29	6.45%
Total	57	103	64	30	5.51%

Table 4: Confusion Matrix – Model MAX22.

	1	2	3	4	Error
1	51	3	0	0	5.56%
2	5	93	0	0	5.10%
3	0	5	65	1	8.45%
4	0	1	2	28	9.68%
Total	56	102	67	29	6.69%

Table 5: Confusion Matrix – Model MIN42.

	1	2	3	4	Error
1	54	0	0	0	0.00%
2	7	91	0	0	7.14%
3	2	4	63	2	11.27%
4	0	0	0	31	0.00%
Total	63	95	63	33	5.91%

Table 6: Confusion Matrix - Model AVG42.

	1	2	3	4	Error
1	54	0	0	0	0.00%
2	7	91	0	0	7.14%
3	2	2	66	1	7.04%
4	0	0	0	31	0.00%
Total	63	93	66	32	4.72%

Table 7: Confusion Matrix – Model MAX42.

	1	2	3	4	Error
1	51	3	0	0	5.56%
2	7	91	0	0	7.14%
3	1	4	66	0	7.04%
4	0	0	0	31	0.00%
Total	59	98	66	31	5.91%

Table 8: Confusion Matrix – Model MIN62.

	1	2	3	4	Error
1	54	0	0	0	0.00%
2	7	91	0	0	7.14%
3	0	1	69	1	2.82%
4	0	0	0	31	0.00%
Total	61	92	69	32	3.54%

Table 9: Confusion Matrix - Model AVG62.

	1	2	3	4	Error
1	54	0	0	0	0.00%
2	7	91	0	0	7.14%
3	2	2	66	1	7.04%
4	0	0	0	31	0.00%
Total	63	93	66	32	4.72%

Table 10: Confusion Matrix - Model MAX62.

	1	2	3	4	Error
1	54	0	0	0	0.00%
2	7	91	0	0	7.14%
3	2	2	67	0	5.63%
4	0	0	1	30	3.23%
Total	63	93	68	30	4.72%

Table 11: Confusion Matrix – Model MIN82.

	1	2	3	4	Error
1	54	0	0	0	0.00%
2	7	91	0	0	7.14%
3	2	1	66	2	7.04%
4	0	0	0	31	0.00%
Total	63	92	66	33	4.72%

Table 12: Confusion Matrix – Model AVG82.

	1	2	3	4	Error
1	54	0	0	0	0.00%
2	8	90	0	0	8.16%
3	3	3	63	2	11.27%
4	0	0	0	31	0.00%
Total	65	93	63	33	6.30%

Table 13: Confusion Matrix - Model MAX82.

	1	2	3	4	Error
1	54	0	0	0	0.00%
2	8	90	0	0	8.16%
3	2	2	67	0	5.63%
4	0	0	0	31	0.00%
Total	64	92	67	31	4.72%

Table 14: Confusion Matrix - Model MIN102.

	1	2	3	4	Error
1	54	0	0	0	0.00%
2	7	91	0	0	7.14%
3	0	5	64	2	9.86%
4	0	0	0	31	0.00%
Total	61	96	64	33	5.51%

Table 15: Confusion Matrix - Model AVG102.

	1	2	3	4	Error
1	54	0	0	0	0.00%
2	8	90	0	0	8.16%
3	2	3	63	3	11.27%
4	0	0	0	31	0.00%
Total	64	93	63	34	6.30%

Table 16: Confusion Matrix - Model MAX102.

	1	2	3	4	Error
1	53	0	1	0	1.85%
2	7	91	0	0	7.14%
3	2	2	65	2	8.45%
4	0	0	0	31	0.00%
Total	62	93	66	33	5.51%

Table 17: Confusion Matrix - Model MIN44.

	1	2	3	4	Error
1	114	0	0	0	0.00%
2	17	174	1	0	9.38%
3	2	3	127	7	8.63%
4	0	1	0	62	1.59%
Total	133	178	128	69	6.10%

Table 18: Confusion Matrix - Model AVG44.

	1	2	3	4	Error
1	113	1	0	0	0.88%
2	20	169	3	0	11.98%
3	3	1	129	6	7.19%
4	0	0	0	63	0.00%
Total	136	171	132	69	6.69%

Table 19: Confusion Matrix – Model MAX44.

	1	2	3	4	Error
1	114	0	0	0	0.00%
2	26	165	1	0	14.06%
3	2	1	129	7	7.19%
4	0	4	2	57	9.52%
Total	142	170	132	64	8.46%

Table 20: Confusion Matrix – Model MIN66.

	1	2	3	4	Error
1	174	3	0	0	1.69%
2	30	239	12	0	14.95%
3	3	27	163	14	21.26%
4	0	5	11	81	16.49%
Total	207	274	186	95	13.78%

Table 21: Confusion Matrix - Model AVG66.

	1	2	3	4	Error
1	177	0	0	0	0.00%
2	31	242	8	0	13.88%
3	4	21	167	15	19.32%
4	0	3	11	83	14.43%
Total	212	266	186	98	12.20%

Table 22: Confusion Matrix - Model MAX66.

	1	2	3	4	Error
1	176	0	1	0	0.56%
2	31	242	8	0	13.88%
3	3	22	169	13	18.36%
4	0	4	10	83	14.43%
Total	210	268	188	96	12.07%

Table 23: Confusion Matrix – Model MIN88.

	1	2	3	4	Error
1	233	9	0	0	3.72%
2	52	307	13	0	17.47%
3	15	38	196	22	27.68%
4	0	6	24	101	22.90%
Total	300	360	233	123	17.62%

Table 24: Confusion Matrix – Model AVG88.

	1	2	3	4	Error
1	233	9	0	0	3.72%
2	52	308	12	0	17.20%
3	9	44	199	19	26.57%
4	1	4	24	102	22.14%
Total	295	365	235	121	17.13%

Table 25: Confusion Matrix - Model MAX88.

	1	2	3	4	Error
1	235	7	0	0	2.89%
2	52	312	8	0	16.13%
3	8	51	187	25	31.00%
4	0	5	24	102	22.14%
Total	295	375	219	127	17.72%

Table 26: Confusion Matrix – Model RF1010.

	1	2	3	4	Error
1	296	12	0	0	3.90%
2	60	381	21	1	17.71%
3	14	71	208	44	38.28%
4	11	1	27	123	24.07%
Total	381	465	256	168	20.63%

Table 27: Confusion Matrix - Model TO1.

	1	2	3	4	Error
1	48	6	0	0	11.11%
2	13	73	11	1	25.51%
3	6	18	35	12	50.70%
4	1	0	7	23	25.81%
Total	68	97	53	36	29.53%

Table 28: Confusion Matrix - Model TO2.

	1	2	3	4	Error
1	50	4	0	0	7.41%
2	17	71	10	0	27.55%
3	5	21	36	9	49.30%
4	1	0	12	18	41.94%
Total	73	96	58	27	31.10%

Table 29: Confusion Matrix - Model TO3.

	1	2	3	4	Error
1	52	2	0	0	3.70%
2	16	67	15	0	31.63%
3	4	20	41	6	42.25%
4	1	0	8	22	29.03%
Total	73	89	64	28	28.35%

Table 30: Confusion Matrix – Model FO1.

	1	2	3	4	Error
1	43	9	2	0	20.37%
2	17	67	11	3	31.63%
3	0	27	36	8	49.30%
4	0	7	4	20	35.48%
Total	60	110	53	31	34.65%

Table 31: Confusion Matrix - Model FO2.

	1	2	3	4	Error
1	88	22	4	0	22.81%
2	38	124	15	15	35.42%
3	7	40	50	42	64.03%
4	4	9	5	45	28.57%
Total	137	195	74	102	39.57%

Table 32: Confusion Matrix - Model FO3.

	1	2	3	4	Error
1	135	36	6	0	23.73%
2	58	142	53	28	49.47%
3	8	68	101	30	51.21%
4	0	9	18	70	27.84%
Total	201	255	178	128	41.21%

Table 33: Confusion Matrix - Model FO4.

	1	2	3	4	Error
1	189	53	0	0	21.90%
2	92	210	50	20	43.55%
3	12	80	133	46	50.92%
4	0	22	26	83	36.64%
Total	293	365	209	149	39.47%

Table 34: Confusion Matrix - Model FO5.

	1	2	3	4	Error
1	244	42	5	17	20.78%
2	166	242	11	44	47.73%
3	38	146	94	59	72.11%
4	0	53	26	83	48.77%
Total	448	483	136	203	47.80%

Table 35: Confusion Matrix – Model FOUE1.

	1	2	3	4	Error
1	48	4	2	0	11.11%
2	18	71	5	4	27.55%
3	4	25	32	10	54.93%
4	2	4	4	21	32.26%
Total	72	104	43	35	32.28%

Table 36: Confusion Matrix – Model FOUE2.

	1	2	3	4	Error
1	44	10	0	0	18.52%
2	23	63	12	0	35.71%
3	0	27	36	8	49.30%
4	0	10	4	17	45.16%
Total	67	110	52	25	37.01%

Table 37: Confusion Matrix – Model FOUE3.

	1	2	3	4	Error
1	38	14	2	0	29.63%
2	26	60	8	4	38.78%
3	0	29	28	14	60.56%
4	2	5	3	21	32.26%
Total	66	108	41	39	42.13%

Table 38: Confusion Matrix – Model FOUE4.

	1	2	3	4	Error
1	44	6	4	0	18.52%
2	22	64	12	0	34.69%
3	2	30	28	11	60.56%
4	0	3	7	21	32.26%
Total	68	103	51	32	38.19%

Table 39: Confusion Matrix - Model FOSH1.

	1	2	3	4	Error
1	47	5	2	0	12.96%
2	22	65	7	4	33.67%
3	3	22	39	7	45.07%
4	2	2	7	20	35.48%
Total	74	94	55	31	32.68%

Table 40: Confusion Matrix – Model FOSH2.

	1	2	3	4	Error
1	44	8	2	0	18.52%
2	24	52	18	4	46.94%
3	1	19	30	21	57.75%
4	0	6	2	23	25.81%
Total	69	85	52	48	41.34%

Table 41: Confusion Matrix - Model FOSH3.

	1	2	3	4	Error
1	44	8	2	0	18.52%
2	18	68	10	2	30.61%
3	2	32	25	12	64.79%
4	0	2	12	17	45.16%
Total	64	110	49	31	39.37%

Table 42: Confusion Matrix - Model FOSH4.

	1	2	3	4	Error
1	44	8	2	0	18.52%
2	17	59	10	12	39.80%
3	2	28	31	10	56.34%
4	0	3	3	25	19.35%
Total	63	98	46	47	37.40%

Appendix B

Link 1: Permanent link to Thesis Repository.

Github: https://github.com/CHNALI006/Masters-Dissertation

Google Drive: https://drive.google.com/drive/folders/1c7rAlVQEda7Pbr89yJKjAcST_nHCbTWi?usp=share_link

Link 2: Permanent link to MICE plots.

Github: https://github.com/CHNALI006/Masters-Dissertation/tree/main/MICE%20plots

Google Drive: https://drive.google.com/drive/folders/1OfqlzfwnmVDkMM0xKyiVUzGDvkcvyibk?usp=share_link

Link 3: Permanent link to R code.

Github: https://github.com/CHNALI006/Masters-Dissertation/tree/main/R%20Scripts%20RF%2C%20MICE%2C%20EDA

Google Drive: https://drive.google.com/drive/folders/1nHWOnfkqQOcFF9ENZ_jVIx8dV--5DhG6?usp=share_link

Link 4: Permanent link to LSTM code.

Github: https://github.com/CHNALI006/Masters-Dissertation/tree/main/Python%20Scripts%20LSTM

Google Drive: https://drive.google.com/drive/folders/1RUvL-m8REp260mjrE31a2trcXV7CN7k8?usp=share_link

Link 5: Permanent link to Datasets used.

Github: https://github.com/CHNALI006/Masters-Dissertation/tree/main/Raw%20Data%2C%20R%20Data%20and%20CSV%20files

Google Drive: https://drive.google.com/drive/folders/1Fj-IWGgji4nXgYj6lA2mdcYRkLe9ah9R?usp=share_link