Analysis of Machine Learning Algorithms for Time Series Prediction

By

Kimendree Naidoo

Submitted in partial fulfilment of the requirements for the degree

MSc Information Technology

Department of Computer Science

University of Cape Town

Supervised By:

Associate Professor D. Moodley

December 2020

Declaration

I, Kimendree Naidoo, hereby declare that the work on which this dissertation/thesis is based, is my original work (except where acknowledgements indicate otherwise) and that neither the whole work nor any part of it has been, is being, or is to be submitted for another degree in this or any other university.

Signed by candidate

Signature: ……

Date: ……

Abstract

Due to the rapidly increasing prominence of Artificial Intelligence in the last decade and the advancements in technology such as processing power and data storage, there has been increased interest in applying machine learning algorithms to time series prediction problems. There are many machine learning algorithms that can be used for time series prediction problems but selecting an algorithm can be challenging due to algorithms not being suitable to all types of datasets. This research investigates and evaluates machine learning algorithms that can be used for time series prediction. Experiments were carried out using the Artificial Neural Network (ANN), Support Vector Regressor (SVR) and Long Short-Term Memory (LSTM) algorithms on eight datasets. An empirical analysis was carried out by applying each machine learning algorithm to the selected datasets. A critical comparison of the algorithm performance was carried out using the Mean Absolute Error (MAE), the Mean Squared Error (MSE), the Root Mean Squared Error (RMSE) and the Mean Absolute Scaled Error (MASE). The second experiment focused on evaluating the stability and robustness of the optimal models identified in the first experiment. The key dataset characteristics identified; were the dataset size, stationarity, trend and seasonality. It was found that the LSTM performed the best for majority of the datasets, due to the algorithm's ability to deal with sequential dependency. The performance of the ANN and SVR were similar for datasets with trend and seasonality, while the LSTM overall proved superior to the aforementioned algorithms. The LSTM outperformed the ANN and SVR due to its ability to handle temporal dependency. However, due to its stochastic nature, the LSTM and ANN algorithms can have poor stability and robustness. In this regard, the LSTM was found to be a more robust algorithm than the ANN and SVR.

Key Words: Time Series, Artificial Neural Network, Support Vector Machine, Long Short-Term Memory

Table of Contents

List of Tables

List of Figures

Glossary

SVR	Support Vector Machine
ANN	Artificial Neural Network
LSTM	Long Short-Term Memory
Machine Learning	A branch of artificial intelligence based on the concept that machines can learn from data and make predictions based on past experience with minimal human assistance
Feature	A characteristic or property of the data being observed
Test data	A portion of the dataset that the algorithm does not see during training, which is used to assess the algorithm's performance
Training data	A portion of the dataset that is used to train and fit the algorithm
Validation data	A portion of the dataset used to assess the algorithm performance while tuning the algorithm's hyperparameters.
Sunspots	Sunspots are a phenomenon on the sun's photosphere where there are spots that appear darker than the surrounding areas. This phenomenon is temporary.
Rossmann's Store	A chain of pharmacies found in countries across Europe.
Canadian Lynx	A species of Lynx found in northern America

Acknowledgements

Throughout this research process, I have received tremendous support and guidance. I would like to thank my supervisor, Deshen Moodley for his guidance and encouragement, enabling me to learn and grow while writing this thesis. A special thanks goes to my husband, for his continuous support and encouragement.

Chapter 1

1 Introduction

1.1 Background

Time series data can be described as data values that are recorded in chronological order [1], [2]. The data values are usually recorded at equal intervals of time. Some examples of time series data include weather measurements, stock prices, sensor data in industrial environments and sales revenue [2], [3], [4], [1]. Over the last few decades, time series analysis and prediction has been a popular research area [5], [1]. Time series analysis involves identifying relationships between variables and extracting meaningful insights. In time series prediction, future data points are forecasted using data from a prior time period. For example, predicting electricity usage for a future time period is one application of time series prediction [1], [6].

There are several traditional time series prediction algorithms, such as linear regression, that have been in use for several years. However, due to the rapidly increasing prominence of artificial intelligence (AI) over the past decade, the use of machine learning algorithms for time series prediction problems in research has seen a rapid growth [7], [8]. Assumptions about the underlying structure of the data are not required for machine learning algorithms in comparison to traditional time series prediction algorithms. Machine learning is a branch of artificial intelligence where machines are used to detect patterns in data and to make predictions with minimal assistance from humans. The underlying statistical algorithms that enable machine learning have existed for many years. The use of machine learning has grown significantly in recent years due to the advancements in technology, such as robust deep neural network algorithms and tools, and the increase in processing power and storage capacity, which allows for building prediction models for very large datasets [9]. Industries such as the medical, energy, manufacturing, financial and sales sectors, seek to take advantage of the benefits that machine learning provides [10].

The outcomes of machine learning algorithms are dependent on the characteristics of the dataset. Consequently, a particular machine learning algorithm may not be suitable to all types of data, making the task of selecting the most appropriate algorithm for an application, challenging [11]. There are many factors to consider when comparing and selecting the most appropriate machine learning algorithm for a problem. The most common factor is the algorithm performance, which evaluates how accurately the algorithm predicts the correct outcome. Other factors include the stability and robustness of an algorithm [12]. Robustness is an important property to consider for dynamic time series data, where the trends and patterns may differ across different partitions of the data. In this

case, the robustness of the algorithm indicates how much the performance of an algorithm differs when training and testing the algorithm on different periods in the data set. Robustness may also analyse an algorithm's performance over different datasets with the same features or the same dataset with noise added [13], [14]. Stability is an important property to consider for stochastic algorithms such as neural networks. The outputs of neural networks can vary vastly depending on the seed selected and the random initialisation of weights. The stability of an algorithm indicates how much the model predictions vary over multiple runs.

1.2 Research Aim and Objectives

This research aims to compare three machine learning algorithms that are used for time series prediction, on multiple datasets with the aim of evaluating each algorithm and identifying the dataset characteristics that affect the algorithm performance and robustness.

The specific objectives of this research are:

- Review and analyse machine learning algorithms that are used for time series prediction and identify the most widely used machine learning algorithms and data sets used in the literature
- Identify dataset characteristics that may affect the performance of the machine learning algorithms and determine whether there is a relationship between the machine learning algorithm performance and the dataset characteristics
- Evaluate the performance of the machine learning algorithms compared to related results found in the literature
- Evaluate the stability and robustness of the selected machine learning algorithms

1.3 Tools and Approach

1.3.1 Overall Approach

The most relevant and widely used machine learning algorithms for time series prediction were reviewed and analysed in terms of their strengths and weaknesses. Three of the most predominantly used machine learning algorithms were selected and applied to eight datasets in order to analyse the relationship between model performance and dataset characteristics. The performance of the selected algorithms was compared based on their stability and robustness and the characteristics of the dataset.

1.3.2 Datasets

An empirical analysis was carried out on multiple datasets from different applications, which are available on open-source data repositories such as Kaggle and the UCI Data Repository. The datasets

were selected based on prominent datasets used in the reviewed literature. The following datasets were selected; the Airline Passengers dataset, the Canadian Lynx dataset, the Exchange Rate dataset, the Stock Indices dataset, the Rossmann's Sales dataset, the Sunspots dataset, the GEFCom2012 Load dataset and the GEFCom2012 Wind dataset. A critical analysis of the performance of each algorithm when applied to each dataset is presented.

1.3.3 Algorithms

The outcome of the literature review showed that the Support Vector Regression (SVR), Artificial Neural Network (ANN) and Long Short-Term Memory (LSTM) were found to be the most prominent algorithms used for time series prediction. Consequently, these algorithms were selected for the experiments in this study.

1.3.4 Evaluation

The performance of the three algorithms were also compared to the performance of experiments reported in existing research using standardised machine learning algorithm evaluation methods. The following evaluation methods were used for the experiments conducted; MSE, RMSE, MAE and MASE. The benefits and caveats surrounding the performance of various machine learning algorithms for time series prediction are discussed in the context of the research and experiments conducted.

1.3.5 Tools

The Python programming language and its associated machine learning libraries namely, Pandas, Keras and Scikit-Learn, were used for implementing and comparing the three algorithms.

1.4 Project Significance

While machine learning is used in many different industries, a specific machine learning algorithm may not be well suited to all applications and all types of datasets [3], [11]. The results of this research can be used to better understand the relationship between time series characteristics and the performance of the SVR, ANN and LSTM. In this way it can ease the algorithm selection process by analysing the characteristics for a given dataset and facilitate the development of an optimal prediction model for any given dataset. It can also be used as a benchmark for future work, to compare the identified relationships for other machine learning algorithms.

1.5 Dissertation Outline

A literature review of related work can be found in Chapter 2, while Chapter 3 covers the experimental design. The results of the experiment are presented in Chapter 4 and an analysis and discussion of the

results can be found in Chapter 5. Chapter 6 concludes the dissertation with a summary of the key findings and recommendations on further research in the domain of time series data.

Chapter 2

2 Literature Review

This chapter provides a review of the literature relating to time series prediction. The chapter is structured as follows. Section 1 describes time series including time series characteristics, time series analysis and time series prediction. Section 2 describes supervised machine learning. Section 3 describes the machine learning process, data preprocessing, evaluation metrics and the experimental process used in existing research. Section 4 discusses the predominantly used algorithms for time series data, while Section 5 consists of information on key benchmark datasets. Section 6 highlights the key findings and describes the trends and limitations of the existing literature.

2.1 Time Series

2.1.1 Time Series Characteristics

A time series, X, can be described mathematically [5], [15], [16] by the vector below:

$$X = x(t) \tag{1}$$

where t is the time interval at which the data was recorded, and x(t) is the variable measured at t.

Time series data can be characterised as either univariate or multivariate. The former consists of a single variable while the latter consists of multiple variables that are time dependent [1], [17]. When modelling multivariate datasets, the relationships between the various time dependent variables need to be considered, making it more complex to model [17].

A time series dataset can have several characteristics such as trend, cyclicity, seasonality, linearity and stationarity [6], [18]. A trend is said to occur when the time series data shows an increase or decrease over a long period of time, while seasonality is a variation that occurs at a certain time of the year, such as an increase of sales during the Christmas period [1], [18], [6]. Cyclic variations have varying frequency and are generally an impact of economic conditions and financial cycles. Cyclicity differs from seasonality in the sense that the former occurs at an unfixed frequency while the latter occurs at a fixed and known frequency [6], [18]. Noise is a component of time series which is considered as a variation of the data that has no pattern and occurs randomly.

A time series can be categorised as linear or non-linear. In a linear time series, a data point can be modelled as a linear function of the past values and the present value, while a non-linear time series can be modelled by a non-linear function [1]. The type of algorithm used on a time series is highly dependent on whether the time series is linear or non-linear.

The stationarity of time series data can be classified as stationary or non-stationary. A time series is stationary when all the statistical properties such as the mean, variance, and standard deviation, remain constant through time; and there is no trend, cyclicity, or seasonality present [1], [4].

2.1.2 Time Series Analysis and Time Series Prediction

There are two aspects to dealing with time series data, namely time series analysis and time series prediction [1]. Time series analysis involves the use of various methods, such as modelling, that allows us to better understand the data, by exploring the relationships between variables and extracting meaningful insights from the data [1]. Time series prediction involves using various algorithms that use past values in the dataset to determine the value at a future time [19], [6], [18].

The types of algorithms that can be used in time series analysis include regression algorithms, time-domain algorithms and frequency-domain algorithms [1]. Some of the simplest prediction algorithms include the naïve method and the average method. The naïve method equates any future values to the last observed value of the dataset. The average method uses the average of the previous dataset values to calculate future values [18]. Linear regression and the ARIMA are some of the other algorithms that can be used for time series prediction [1], [18], [4].

Regression is a commonly used method that is based on estimating and understanding the relationships that occur between two or more variables [1]. There are various types of regression algorithms that can be applied to a prediction problem such as simple regression, multiple regression and non-linear regression [1], [18]. The Box Jenkins algorithm highlights a universal algorithm that can be applied to a time series prediction problem which entails the use of an Autoregressive algorithm [1].

The Autoregressive Moving Average (ARMA) algorithm is a commonly used traditional statistical algorithm, that consists of two types of time series prediction algorithms, namely the Autoregressive (AR) and Moving Average (MA) algorithms [6], [20], [21]. The AR algorithm makes predictions by using the behavior of past values whereas the MA algorithm makes predictions based on past forecast error values [20], [18]. The ARMA is an algorithm that is used with stationary data. If the data is non-stationary, it must be transformed before applying the ARMA. A non-stationary time series can be converted to a stationary time series using a method called differencing. Differencing calculates the difference between the current record and previous record [18].

The Autoregressive Integrated Moving Average (ARIMA) algorithm is created through the combination of the ARMA and differencing. The ARIMA algorithm is mathematically described by Zhang [21] in the equation below.

$$y_t = \theta_0 + \phi_1 y_{t-1} + \cdots + \phi_a y_{t-a} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \cdots - \theta_b \varepsilon_{t-b} \quad (2)$$

Where:

y_t is the actual value,

ε_t is the random error,

ϕ_i and θ_j are algorithm parameters with (i = 0, 1, 2, …, a) and (j = 0, 1, 2, …, b)

The use of the ARIMA algorithm is limited due to the assumption that the data in question is linear [6], [21]. Since real-world data is rarely linear, variations of ARIMA, various supervised machine learning algorithms, as well as hybrid algorithms have been proposed to overcome this limitation as seen in the reviewed literature [22], [23], [20], [21]. Due to the increased interest in AI, supervised machine learning algorithms have become popular; and have been used extensively in recent research, to determine if its performance is superior to the ARIMA [5], [7].

2.2 Supervised Machine Learning

There are different types of machine learning algorithms, namely supervised, unsupervised and reinforcement learning [23]. Supervised learning requires labelled data whereby the required output is provided in the training data. Conversely, unsupervised learning does not require labelled data. Supervised learning is commonly used for classification and regression while unsupervised learning is used for clustering. Reinforcement learning uses a trial and error approach where feedback is given to the system to determine if an action carried out is correct or not in order to reach a desired goal [9], [24].

In order to evaluate machine learning algorithms, datasets are split into training and test datasets. The training dataset is used to train the model and the test dataset is used to determine how the model performs on out of sample data. A portion of the training data known as a validation set is used to tune algorithm hyperparameters as part of the model training process.

Classification is used to predict discrete output variables while regression is used to predict continuous output variables [9], [25], [24], [26]. Classification algorithms are used to predict the class to which an observation belongs. An example would be the prediction of whether an email received is a social email or a promotional email [9]. Regression algorithms are used to predict continuous variables such as predicting the number of passengers for an airline [9]. There are several machine learning algorithms that can be used either for classification or regression and there are algorithms that can be used for both classification and regression problems. Popular classification algorithms include K-Nearest Neighbor, Decision Trees, Random Forest, Logistic Regression, Naïve Bayes, Multilayer

Perceptron and Support Vector Machine [23]. Machine learning algorithms that can be used for regression problems include K-Nearest Neighbor, Decision Trees, Linear Regression, Support Vector Regression and Random Forest [23]. The studies reviewed predominantly focused on continuous datasets therefore this research has a strict focus on machine learning algorithms for regression scenarios.

2.3 Machine Learning for Time Series Data

2.3.1 Machine Learning Process Overview

The figure below shows the commonly adopted machine learning process followed in the reviewed literature [27], [28].

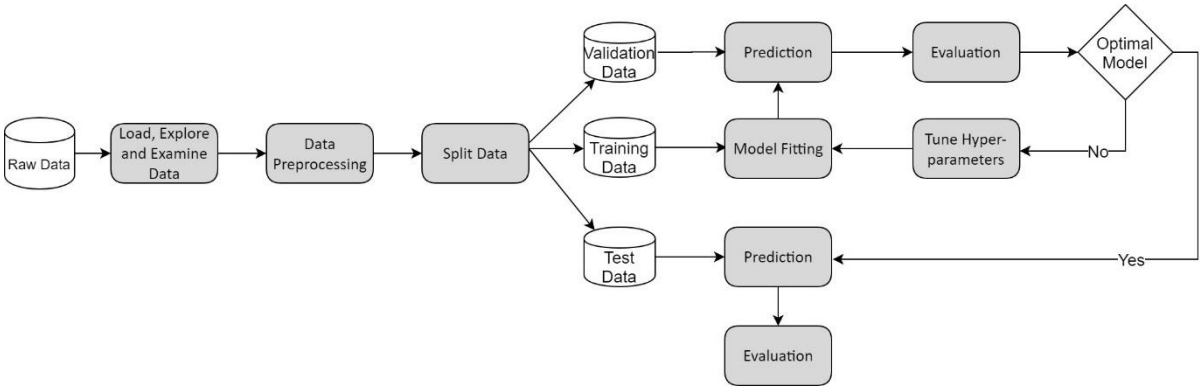

Figure 2-1: Machine Learning Process Diagram

Once the data was loaded in a readable format, it was explored in order to gather basic information and to determine what can be intuitively inferred directly from the data without manipulation. In this phase an understanding of the size of the data, the shape, and statistical characteristics were obtained. Following this phase is data preprocessing which may include transforming the data and dealing with missing values. The data was then split into training, test and validation datasets. A model was fitted to the training data. The validation data was used to determine optimal hyperparameter values. Once acceptable training results were obtained, the optimal model was then used to make predictions on the test data in order to assess the performance of the model on out of sample data.

2.3.2 Data Preprocessing

The preprocessing techniques observed in studies vary based on the characteristics of the dataset and the algorithm being used. A common preprocessing technique was the use of statistical methods, to convert non-stationary data to stationary data by using the log transformation and differencing. Ghiassi et al. [28] and Himakireeti and Vishnu [29] carried out log transformation and differencing for the Airline Passenger dataset as the ARIMA was used in their studies. Khashei and Bijari [30] and Zhang

[21] used the natural logarithm to transform the Exchange Rate data while Masum et al. [16] transformed the data to make it stationary by first applying a log transformation and then carrying out first order differencing.

Normalisation is a common preprocessing technique used to convert numeric data to a common scale. It is important to note that when normalising time series data, the data should be first split into the training and test datasets. If normalisation occurs before partitioning of the data, the test data will be used to calculate the statistics of the training data which can leak information about the test data to the algorithm. Samsudin et al. [27] normalised the datasets within the range [0, 1] in order to accurately compare the results across datasets. The min-max normalisation is often used to scale data between the range [0, 1]. Dingli and Fournier [23] opted to normalise the Exchange Rate dataset and the Stock Indices dataset by using Z-score scaling. Yu et al. [31] converted the Rossmann's Sales dataset into numerical values, removed observations where the store was closed, and normalised the data by subtracting the mean sales from the sale numbers.

Other studies opted to use the original data and did not transform the dataset. Ghiassi et al. [28], Khashei and Bijari [30] and Zhang [21] did not transform the Sunspot dataset. Adhikari and Agrawal [5] used the original Airline Passenger data while implementing the SVR and rescaled data to evaluate the ANN.

A rolling window is used to calculate the dataset statistics such as mean or median over a given number of previous observations. Silva [32] and Charlton and Singleton [33] used a rolling average to smoothen the data.

2.3.3 Dataset Partitioning

Dataset partitioning involves splitting the dataset into a training set and a test set. Dataset partitioning aids in the evaluation of the algorithm, hyperparameter tuning and determining the generalisabilty of the algorithm. The training set is sometimes split further to allow for a validation set, which is used for hyperparameter tuning.

2.3.3.1 Training and Test

Two of the most common traditional dataset partitioning methods include cross validation and the hold out method [34]. The hold out method is one of the basic methods used for splitting data which entails holding a portion of the data for testing while the remainder of the data is used for training. This method was used in the reviewed literature to partition the data into the test and training sets. The hold out method is traditionally used for time series datasets as it handles the temporal

dependency of time series well. This is because the chronological order of the data is kept constant [34].

A common partitioning ratio for the hold out method is the 70:30 percent ratio, which entails using 70 percent of the dataset for training and 30 percent for testing. This has been used by Wang and Li [31] and Yu et al. [26]. Although Samsudin et al. [22] found that the 75:25 percent ratio was popular; the authors chose to use an 85:15 percent ratio. The partitioning varied with each dataset, particularly with the smaller datasets which did not demonstrate any common partitioning ratios. All the reviewed literature that made use of the Canadian Lynx dataset, used 12 percent of the data for testing. Adhikari and Agrawal [9], Ghiassi et al. [23], Khashei and Bijari [25] and Zhang [16] opted to use 77 percent of the Sunspots dataset for training and 23 percent for testing. Adhikari and Agrawal [9], Ghiassi et al. [23] and Khashei and Bijari [25] used 8 percent of the Airline Passenger dataset for training and 92 percent for testing. Khashei and Bijari [25] followed Zhang [16] in using 12 months of data for testing the Exchange Rate dataset.

2.3.3.2 Validation Data for Hyperparameter Tuning

A basic cross validation method is k-fold cross validation, which is commonly used for splitting the training data, to obtain a validation set that is used for hyperparameter tuning of algorithms such as the SVR. This method involves randomly shuffling and splitting the data into equal sized k portions; and is based on the assumption that the data is independent and is distributed identically [34]. The standard k-fold cross validation method is not suitable to time series data as it cannot handle the temporal dependency of the data. A time series should be kept in chronological order to prevent data leakage, where future data points are used to train the algorithm. Since k-fold cross validation shuffles the data and does not keep the data in sequence, it cannot be used on time series data [34]. In the standard k-fold cross validation, a single portion of the data is held out for testing while the remainder of the dataset is used for training. This process is repeated for k iterations such that each of the portions are used for testing.

The hold out method can be shorter to run and requires less computational power. However, cross validation yields a trained algorithm with a higher accuracy than the hold out method [35]. A number of variations of k-fold cross validation have been proposed to overcome its limitations dealing with time series data. Cerqueira et al [34] state that cross validation in blocked form is superior for stationary time series. However, approaches that maintain the temporal order should be used on non-stationary time series. Cerqueira et al [34] discuss a prequential approach where the data is split into blocks, and a block is first used to test the model and then used to train the model in the next iteration.

Adikari and Agrawal [9], Kraus et al [36], Yu et al [31] and Mangolova and Agafonov [37] use variations of cross validation that account for the temporal dependencies of time series data. Samsudin et al. [27] and Ismail and Shabri [38] opted to use the k-fold cross validation and did not discuss the implications of using k-fold cross validation on time series data. The rest of the studies use the hold out method and do not discuss cross validation and the implications of using it on time series data.

2.3.4 Hyperparameter Tuning

Hyperparameter tuning is the process of determining the optimal set of parameters for a machine learning algorithm. Hyperparameters are parameters that are selected prior to the training of an algorithm. Hyperparameter tuning is carried out by trying different parameter combinations to determine what the optimal combination is.

Samsudin et al. [27] carried out hyperparameter tuning for the SVR using the following hyperparameter value ranges; C had the range of [1, 15] in increments of 1, epsilon had a range [0.1, 0.5] with increments of 0.1 and gamma set at 0.5. While Wang and Li [39] found a polynomial kernel to be superior, Samsudin et al. [27] , Adhikari and Agrawal [5] and Ismail and Shabri [38] found the radial basis function kernel (RBF) to have the best performance. The RBF is often used in studies as it allows for the modelling of both linear and non-linear datasets [40]. Ojemakinde [40] also highlights that the computational cost for hyperparameter tuning is lower for the RBF due to the low number of hyperparameters in comparison to other kernels.

Adhikari and Agrawal [5] state that there are no set rules for the selection of hyperparameters of the ANN. Samsudin et al. [27] followed a trial and error approach for the optimisation of the number of hidden neurons and layers for the ANN. The authors used a learning rate of 0.001, the sigmoid transfer function and trained for 5000 epochs.

2.3.5 Evaluation Metrics

Evaluation allows one to determine an algorithm's ability to generalise. Generalisation refers to an algorithm's ability to use the patterns learnt previously to adapt to new, unseen data. Overfitting and underfitting are issues that are associated to generalisation and can lead to poor performance with machine learning algorithms. Underfitting occurs when an algorithm models the training data poorly and is unable to generalise to unseen data. Overfitting occurs when an algorithm models the training data too well and the performance of the model is substantially worse on out of sample data in comparison to the training data.

Some of the evaluation metrics that have been used in the reviewed literature include Mean Squared Error (MSE), R-Squared, Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE),

Mean Absolute Error (MAE), Mean Absolute Scaled Error (MASE), Theil's U Statistics, relative RMSE, Root Mean Square (RMS), Normalized MSE (NMSE), Weighted RMSE (WRMSE) and Root Mean Square Percentage Error (RMSPE). The most common evaluation metrics used in the reviewed literature include the MSE and RMSE. Research involving the Airline dataset, Sunspot dataset, Canadian Lynx dataset, and Exchange Rate dataset commonly use the MSE, while for the Stock Indices and Wind datasets the RMSE is predominantly used. Research involving the Rossmann's dataset typically uses the RMSE and research with the Load dataset uses an RMSE and weighted RMSE.

The MSE measures the average squared difference between the forecasted and actual values and can be mathematically expressed as illustrated in the following equation [5], [38].

$$\mathrm{MSE} = \frac{1}{\mathrm{n}}\sum_{\mathrm{t}=1}^{\mathrm{n}}(\mathrm{y_t} - \mathrm{p_t})^2 \tag{3}$$

Where, y_t is the actual value and p_t is the predicted value.

The MSE is more sensitive to large errors than the other evaluation metrics since the errors are squared which gives an uneven weight to extremely large errors [5]. Adhikari and Agrawal [5] highlight that the MSE provides a fairly good indication of the error between the predicted and actual values. However, this metric is not easily interpreted as compared to other metrics such as the MAPE and RMSE as it is in the squared form of the data's scale.

The RMSE is essentially the square root of the MSE and can be mathematically formulated as seen in Equation (4) [5], [6], [27], [32].

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{n}}\sum_{\mathrm{t}=1}^{\mathrm{n}}(\mathrm{y_t} - \mathrm{p_t})^2} \tag{4}$$

Chatfield [6] reiterates that the RMSE is easier to interpret compared to the MSE as it is measured in the same scale as the data. The RMSE is not well suited to comparing performance for different datasets with data of different scales, as it is unit dependent.

RMSPE is the percentage of RMSE and is given in Equation (5). The RMSPE is better suited to comparing performance across datasets with different units as it is not scale dependent.

$$\mathrm{RMSPE} = \sqrt{\frac{\frac{1}{\mathrm{n}}\sum_{\mathrm{t}=1}^{\mathrm{n}}(\mathrm{y_t} - \mathrm{p_t})^2}{\mathrm{y_t}}} \times 100 \tag{5}$$

The MAE measures the absolute average difference between the forecasted and actual values. It can be represented mathematically as seen in the equation below [5], [6], [27]. The MAE is scale dependent and has the same drawbacks as the RMSE.

$$\mathrm{MAE} = \frac{1}{\mathrm{n}}\sum_{\mathrm{t}=1}^{\mathrm{n}}|\mathrm{y_t} - \mathrm{p_t}| \tag{6}$$

The MAPE is essentially the percentage of the MAE and therefore measures the percentage of the absolute difference between the forecasted and actual values. The MAPE is not scale dependent. The MAPE is expressed mathematically by the following equation [5].

$$\mathrm{MAPE} = \frac{1}{\mathrm{n}}\sum_{\mathrm{t}=1}^{\mathrm{n}}\left|\frac{\mathrm{y_t} - \mathrm{p_t}}{\mathrm{y_t}}\right| \times 100 \tag{7}$$

The MASE is the MAE divided by the training MAE of the naïve forecast. It can be depicted mathematically by the following equation [18].

$$\mathrm{MASE} = \frac{\frac{1}{\mathrm{n}}\sum_{\mathrm{t}=1}^{\mathrm{n}}|\mathrm{y_t} - \mathrm{p_t}|}{\frac{1}{T-m}\sum_{\mathrm{t}=\mathrm{m}+1}^{T}|\mathrm{y_t} - \mathrm{y_{t-m}}|} \tag{8}$$

m is the seasonality and is 1 if there is no seasonality.

In order to compare the performance of machine learning models across a number of datasets, an evaluation metric that is not scale dependent should be used. The RMSPE and MAPE are not unit-specific and can be used to evaluate the performance of algorithms across multiple datasets. The disadvantage of percentage evaluation metrics is that they will be undefined or infinite if the actual value is zero or close to zero. The MASE was not used in any of the benchmark studies. However, Hyndman and Koehler [41] highlighted that the RMSPE and MAPE are not reliable evaluation metrics and recommended a Mean Absolute Scaled Error (MASE) to overcome the deficiencies of the aforementioned metrics.

2.3.6 Experimental Process

For studies carried out by Aladag et al [22], Ghiassi et al [28], and Khashei et al [30], where the results of applying a proposed algorithm to several datasets were compared to existing research, the experimental process consisted of the machine learning process described in Section 2.3.1, being applied for each dataset. Adhikari and Agrawal [5] carried out different preprocessing methods on each dataset for every algorithm used. Conversely, Samsudin et al. [27] only preprocessed each dataset once. The authors then compared the results of each dataset/algorithm combination.

Research that had multiple prediction steps such as research involving the Sunspot dataset, had additional process steps for the prediction phase using the test data. Yu et al. [31] trained each algorithm on data obtained from a single store and thereafter made predictions for the store in question. Silva [32] carried out data preprocessing, algorithm training and post processing in a sequential manner. The preprocessing consisted of feature extraction followed by clustering of features. The algorithm training included algorithms for each wind farm of the Wind dataset, which were trained for a subset of the data and thereafter for all observations. The post processing included an ensemble algorithm followed by smoothing and predictions.

2.4 Machine Learning Algorithms

There are numerous machine learning algorithms that can be used for time series prediction. However, selecting the most appropriate algorithm can be a challenging task. Machine learning algorithms are not always well suited to all types of data. Adhikari and Agrawal [5] have compared various traditional statistical prediction algorithms with some machine learning algorithms, for datasets with different characteristics. The machine learning algorithms include the ANN and SVR, but the study lacks the use of an LSTM. Samsudin et al. [27] evaluated the performance of the SVR and ANN on various datasets that have different statistical characteristics. However, this research also lacks the evaluation of recurrent neural networks, specifically the LSTM which Kraus et al. [36], Borovykh et al. [42] and He [43], found to be the most appropriate for time series data.

A number of studies involved the comparison of traditional statistical algorithms, the ANN and SVR with proposed hybrid algorithms. Zhang [21] proposed an RNN-ARIMA hybrid which was used as a benchmark by Aladag et al. [22], Ghiassi et al. [28], Ismail and Shabri [38] as well as Khashei and Bijari [30] for their own proposed hybrid algorithms. Gumani et al. [20] studied various machine learning algorithms and hybrids with the focus on a single dataset. The LSTM was explored by Kraus et al. [36] on a single dataset. The algorithms that were most commonly used in the reviewed literature was the ARIMA, SVR, ANN and various hybrids.

Since this research has a strict focus on machine learning algorithms in its simplest form, the ANN and SVR were selected to be explored. The LSTM was also selected as it is thought to be well suited to time series data.

2.4.1 Support Vector Regressor

Support Vector Machine (SVM) is a supervised learning algorithm that can be applied in classification and regression scenarios [19], [44]. The form of the SVM algorithm that is applied to regression problems is often called Support Vector Regression (SVR) [45], [1], [46], [19].

The SVM algorithm is often used for classification problems by determining the optimal hyperplane that differentiates the classes. The hyperplane occurs in a multidimensional space where the number of dimensions equate to the number of features [1], [9], [47]. The hyperplane acts as a boundary line that categorises a data point into one of the classes. The data points that are found near the hyperplane are called Support Vectors. The location of the hyperplane is dependent on the Support Vectors as the objective is to maximise the space between the hyperplane and its nearest data point [1], [47].

In the event of linear classification, a low dimensional space is sufficient. However, if data cannot be linearly separated, a higher dimensional space is required. This is achieved with the use of functions called Kernels which transform the feature space into a high dimensional space [1], [48], [47]. Based on studies by Ajoy and Dobrivoje [1] and Shin et al [47], adapted from Vapnik's research [49], [50], the SVM can be described mathematically as follows.

Consider a training set of data (x_i , y_i) where i = 1, 2, …., M, x_i is the training input and y_i is the corresponding training output data. For data that is linearly distinct the hyperplane can be described with the following equation.

$$\mathrm{W^T x_i} + \mathrm{c} = 0 \tag{9}$$

Where, W^T is a weight vector, and c is a scaler.

A Kernel function can be described mathematically as follows while a hyperplane for a dataset that is non-linear is described in Equation (11).

$$k(x) = [k_1(x), k_2(x) \dots k_m(x)] \tag{10}$$

$$\sum_{i=1}^{M} w_i x_i + c = 0 \tag{11}$$

The objective of the SVM is to find the optimal hyperplane by identifying the optimal values of c and W, which maximises the hyperplane between the classes. In order to achieve the optimal W, the cost function is minimised using Lagrange multipliers. The derived equation for the optimal W is demonstrated in Equation (12).

$$W = \sum_{i=1}^{M} \lambda_i y_i k(x_i) \tag{12}$$

Where, $k(x_i)$ is a feature vector that corresponds to the input vector.

The output of the SVM for a dataset that can be linearly classified is shown in the equation below.

$$y(x) = sgn\left(\sum_{i=1}^{M} y_i \lambda_i (x_i \cdot x) + c\right) \tag{13}$$

When taking into consideration the Kernel function for a dataset is not linear, the output is generated with Equation (14). The Kernel function produces the inner product of the support vector and input data.

$$y(x) = sgn\left(\sum_{i=1}^{M} y_i \lambda_i K(x_i, x) + c\right) \tag{14}$$

There are numerous kernel functions such as Polynomial Kernels, Radial Bias Function (RBF) Kernels and Sigmoid Kernels which are described in Equations (15), (16), and (17) respectively [1], [47].

$$K(x, x_i) = \big((x, x_i) + 1\big)^a \tag{15}$$

Where a is the polynomial degree.

$$K(x, x_i) = e^{\frac{-|x-x_i|^2}{2\beta^2}} \tag{16}$$

Where, β^2 is the bandwidth of the Radial Bias Function.

$$K(x, x_i) = S[(x, x_i)] \tag{17}$$

The principles that the SVM is based on also forms the basis for the SVR. However, it becomes more complex as the SVR involves continuous outputs. The outputs of the SVR have an infinite number of possibilities whereas the SVM outputs are confined to the number of classes.

The objective of the SVR is to minimise the error function given in Equation (18) [45]. Consider a training dataset (x_i , y_i) where i = 1, 2, …., M, x_i is the training input and y_i is the corresponding training output data.

$$J = \frac{1}{2}\big||W|\big|^2 + A\sum_{i=1}^{M} |y_i - f(x_i)|_\epsilon \tag{18}$$

Where W is the weights and A is a constant.

The linear function that minimises the loss function is described in Equation (19) while Equation (20) includes the Kernel function that is used for data that is non-linearly separable [45], [19].

$$y(x) = \sum_{i=1}^{M} (\alpha_i^* - \alpha_i) x_i x + c \tag{19}$$

$$y(x) = \sum_{i=1}^{M} (\alpha_i^* - \alpha_i) K(x_i x) + c \tag{20}$$

2.4.2 Artificial Neural Network

In recent years the Artificial Neural Network (ANN) has become an alternative to traditional time series prediction algorithms. The ANN is a machine learning algorithm that is very loosely based on the human brain's ability to learn and comprehend data. The ANN has the ability to take inputs, train itself to detect and learn from patterns and make predictions on new data [5], [6], [20], [27], [21]. The ANN can be applied to non-linear data and its ability to process data in parallel allows for accurate results [21]. Kraus et al. [36] state that the most commonly used ANN algorithms include the Multi-layer perceptron (MLP), the Convolutional neural network (CNN), Long short-term memory (LSTM) and the Gated recurrent unit (GRU). The MLP and CNN are feedforward neural networks while the LSTM and GRU are recurrent neural networks. Feedforward neural networks do not consist of a feedback loop and signals only run from input to output. In recurrent neural networks signals can run in both directions and consist of a feedback loop [51].

2.4.3 Multi-Layer Perceptron

An MLP consists of a minimum of three layers, an input layer, a hidden layer, and an output layer. The first layer which is the input layer, receives the data. The functionality of the hidden layer is to identify the relationships and patterns in the data; while the output layer provides the algorithm's predictions [5], [20], [27], [21]. Each layer consists of a group of neurons which is a mathematical function that imitates the neuron functionality of a brain. The neurons in the neighbouring layers are connected to each other via channels that have weights. Weights indicate of the strength of the relationship between two neurons [15]. The output of a neuron's calculation is used as an input to the other neurons connected to it. The calculation occurring within the neuron is carried out in two phases namely the aggregation and activation phases. The aggregation function multiples the inputs by its weights and sums the output which is then fed as an input to the activation function [15].

The figure below shows the architecture of the MLP, where $x_1, x_2 \ \ldots \ x_i$ is the time series and y_t is the predicted output at a given time.

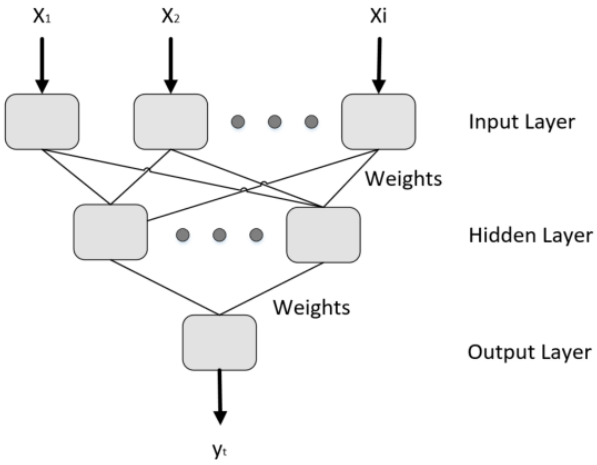

Figure 2-2: Architecture of ANN

The MLP can be expressed mathematically as seen below [5], [27], [21].

$$y_t = \delta_0 + \sum_{j=1}^{a} \delta_j g(\omega_{0j} + \sum_{i=i}^{b} \omega_{ij} y_{t-i}) + \varepsilon_t \tag{21}$$

Where, δ_j and ω_{ij} are the weights with (j = 0, 1, 2, …, a) and (i = 0, 1, 2, …, b). The number of input and hidden neurons are represented by a and b respectively. The random shock is represented by ε_t, δ_0 and ω_{0j} are the terms representing bias, and g represents the activation function. The most common activation functions that are used include the logistic sigmoid function, the rectified linear unit function (relu), and the hyperbolic tangent function, which are represented mathematically in Equations (22), (23) and (24) respectively [36], [27].

$$g(x) = \frac{1}{1 + e^{-x}} \tag{22}$$

$$g(x) = \max(0, x) \tag{23}$$

$$g(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{24}$$

2.4.4 Recurrent Neural Network

Recurrent neural networks (RNN) are artificial neural networks that consist of feedback loops which allow the network to have access to all the past data points [42], [15]. This ability allows the RNN to overcome the limitation of traditional machine learning algorithms, where the network is only able to

access a fixed number of data points. Unlike the MLP, the RNN has two hidden layer inputs namely the present value and past value [36], [52]. The previous data points are continuously stored in the connection between neurons, called a state [36], [51]. A drawback to the RNN is the vanishing gradients problem; when the gradients of the weights reach extremities that are too high or low caused by a high number of feedback loops in long data sequences [53], [15], [43], [36]. Kraus et al. [36] and He [43] state that the limitations of the basic RNN can be overcome with the use of the Long Short-Term Memory (LSTM) algorithm. Figure 2-3 demonstrates the architecture of the RNN.

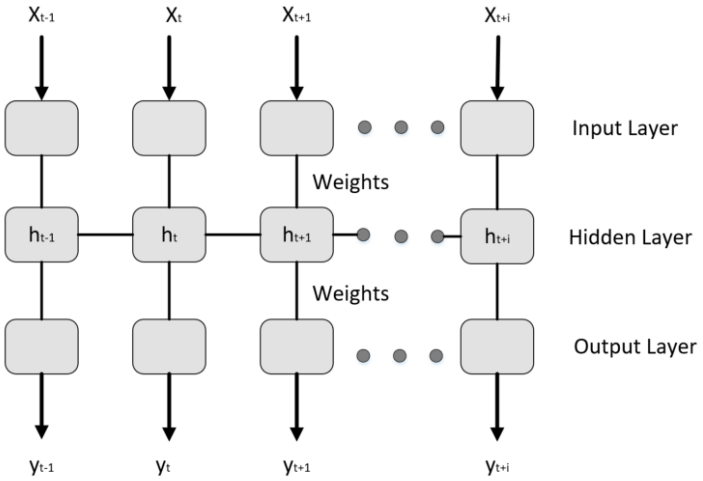

Figure 2-3: Architecture of RNN

Consider a sequence dataset $x = x_1, x_2, \dots, x_t$, the basic RNN can be expressed mathematically as seen in the equation below [54].

$$\mathrm{y_t} = \mathrm{W_{hy}h_t} + \mathrm{c_y} \tag{25}$$

Where W and c are the algorithm's parameters, namely the weights and bias respectively. The hidden layer can be expressed mathematically as seen below [54].

$$\mathrm{h_t} = \mathrm{f}(\mathrm{W_{xh}x_t} + \mathrm{W_{hh}h_{t-1}} + \mathrm{c_h}) \tag{26}$$

Where f is the hidden layer activation function.

2.4.5 Long Short-Term Memory

The LSTM builds upon the RNN by having additional units in the hidden layer containing memory cells and gates to store information and control the flow of information respectively. The gates include input, output and forget gates, which serve to control the input flow, output flow and the resetting of the unit respectively [36], [52]. Figure 2-4 depicts the memory unit of the LSTM algorithm.

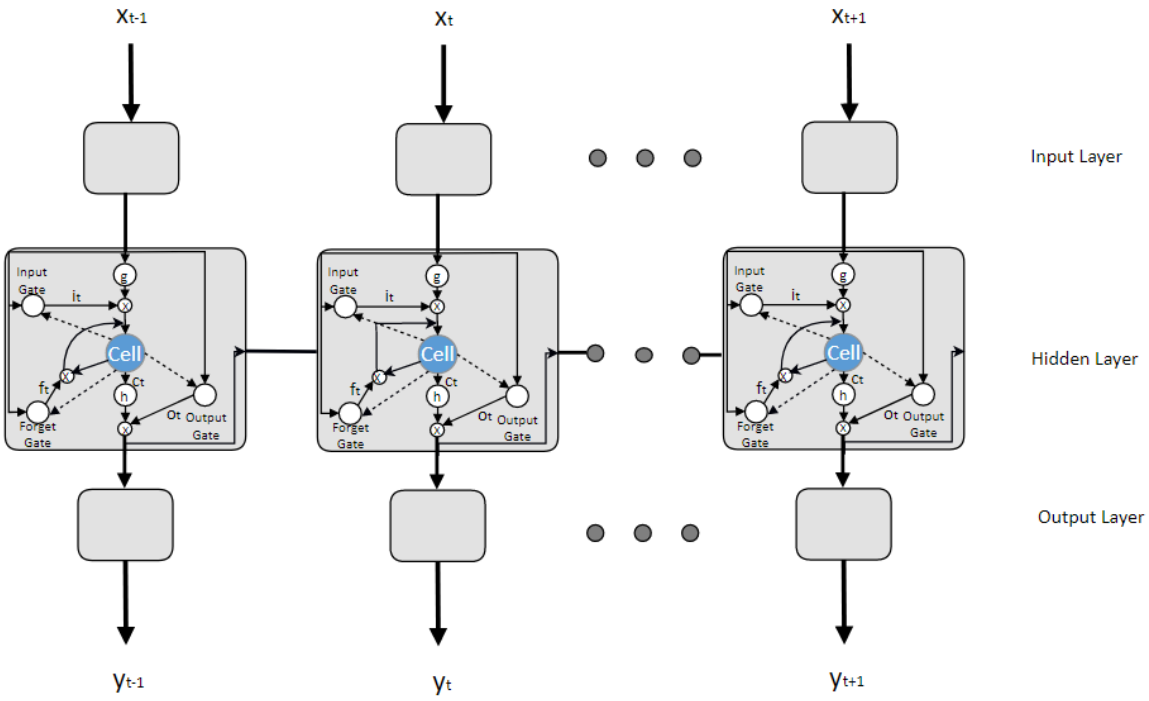

Figure 2-4: Architecture of LSTM

The LSTM can be expressed mathematically by the following equations [52]:

$$i_t = \sigma(W_{ix}x_t + W_{im}(o_{t-1} \cdot h(c_{t-1})) + W_{ic}c_{t-1} + b_i) \quad (27)$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}(o_{t-1} \cdot h(c_{t-1})) + W_{fc}c_{t-1} + b_f) \quad (28)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g(W_{cx}x_t + W_{cm}(o_{t-1} \cdot h(c_{t-1})) + b_c) \quad (29)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}(o_{t-1} \cdot h(c_{t-1})) + W_{oc}c_t + b_o) \quad (30)$$

$$y_t = \emptyset(W_{ym}(o_t \cdot h(c_t) + b_y) \quad (31)$$

Where W represents the weights while b depicts the bias. The logistic sigmoid function is represented by σ. The input, forget, and output gates are represented by i, f and o respectively. The cell activation functions are depicted by g and h while $\emptyset$ is softmax which is the network activation function.

2.5 Machine Learning Applications

It was noted that in the reviewed literature, there are common themes regarding the type of datasets used for analysing the performance of machine learning algorithms. The reviewed research has a focus on common applications and industries with some of the popular dataset types being stock index prices, exchange rates, business sales, electricity consumption, and weather. The review of related

work on the application of machine learning algorithms is restricted to studies performed on eight of the most commonly used datasets. These datasets vary in the type of applications they are used in, which include business, finance, environmental and energy applications. The selected datasets are the S&P500 Stock Index, the British Pound/US Dollar Exchange Rate, the Airline Passengers dataset, the Sunspot dataset, the Rossmann's Sales dataset, the Canadian Lynx dataset, the Global Energy Prediction Competition (GEFCom2012) Load dataset, and the GEFCom2012 Wind dataset.

2.5.1 Airline Passengers Dataset

The Airline Passenger dataset is a time series that captures the number of passengers an international airline accommodates monthly, over a time period between January 1949 and December 1960 [55]. It is a small dataset with only 144 observations [5], [28], [30], [27], [29]. This dataset has multiplicative seasonality that's subject to an increasing trend [5], [27], [30], [28]. The dataset is non-stationary which was confirmed with the use of stationarity tests [5], [30], [28], [29]. The dataset does not show any linearity [30], [28]. Various experiments were carried out using this dataset including a comparison between the SVR and ANN by Samsudin et al. [27], while Adhikari and Agrawal [5] opted to compare the ARIMA, ANN, Seasonal ANN and SVR. The results conflicted as the former showed that the ANN performed better while the latter had best performance for the SVR.

A summary of the Airline dataset experiment results for the reviewed literature can be seen in Table 2-1.

Table 2-1: Experiment and Results Summary for Airline Dataset

Paper	**Algorithms**	**Results**		
		MSE	RMSE	MAPE
[5]	SARIMA	189.333893	13.759865	2.244234%
	ANN	248.794863	15.773232	3.025739%
	SANN	275.720525	16.604834	2.486088%
	SVR	**176.885301**	**13.299823**	**2.336608%**
[28]	ANN	0.29		
	ARIMA	0.43		
	DAN2	**0.19**		
[29]	ARIMA	Identified optimal parameters for ARIMA		
[27]	SVR		0.0866	

Paper	Algorithms	Results		
	FFNN		**0.0693**	

2.5.2 Sunspot Dataset

The Sunspot dataset is a time series that captures the yearly average number of sunspots observed between the period of January 1700 and January 1987. The dataset consists of 288 observations [5], [28], [30], [21]. The dataset is found to be non-stationary and non-linear [5], [30], [28], [21]. It was also noted that the data was non-gaussian and had a cyclic pattern with a mean of 11 years [21], [30].

A various number of experiments have been carried out on this dataset including comparisons between the AR, ANN and SVR [5] as well as comparisons between machine learning algorithms in its simplest form and hybrid algorithms [21], [28], [30]. Table 2-2 presents a summary of the experiments carried out as well as the results.

Table 2-2: Experiment and Results Summary for Sunspot Dataset

Paper	Algorithms	Results		
		MSE	**RMSE**	**MAPE**
[5]	AR	483.561260	21.990026	60.042080%
	ANN	**334.173011**	**18.280400**	**30.498342%**
	SVR	792.961254	28.159568	40.433136%
[28]	Compared with Zhang [21]			
	DAN2	**258**		
[30]	Compared with Zhang [21]			
	Proposed Algorithm	**234.206103**		
[27]	SVR		0.0777	
	FFNN		**0.0711**	
[21]	ANN	351.19366		
	ARIMA	306.08217		
	Proposed Algorithm	**280.15956**		

2.5.3 Canadian Lynx Dataset

The Canadian Lynx dataset consists of 114 records and captures the quantity of lynx trapped annually in Northern Canada during the period between 1821 and 1934 [56]. The dataset is non-stationary [5], [21], [28], [30]. A number of studies were carried out on this dataset some of which included the comparison between machine learning algorithms in its basic form while others opted to propose hybrid algorithms that outperformed basic algorithms [5], [21], [22] ,[57], [38]. The table below gives a summary of the experiments and results in the reviewed literature.

Table 2-3: Experiment and Results Summary for Canadian Lynx Dataset

Paper	**Algorithms**	**Results**		
		MSE	**RMSE**	**MAPE**
[5]	AR	**0.005123**	**0.071577**	1.950160%
	ARMA	0.016533	0.128581	**3.409039%**
	ANN	0.012659	0.112512	**2.392407%**
	SVR	0.052676	0.229513	5.811812%
[22]	Compared with Zhang [21], Kajitani [57]			
	FFNN	0.020		
	Proposed Algorithm	**0.009**		
[28]	Compared with Zhang [21]			
	DAN2	**0.006**		
[30]	Compared with Zhang [21]			
	Proposed Algorithm	**0.006**		
[57]	AR		0.134	
	FFNN		0.129	
	SETAR		**0.122**	
[21]	ANN	0.020466		
	ARIMA	0.020486		
	Proposed Algorithm	**0.017233**		

Paper	Algorithms	Results		
[38]	Compared with Zhang [21], Kajitani [57], Khashei [30], Aladag [22]			
	SVR	0.0085		
	LSSVR	**0.0030**		

2.5.4 Rossmann's Sales Dataset

The Rossmann's Sales dataset includes the daily sales data for 1115 of the Rossmann's stores over the period 1 August 2015 to 17 September 2015 [55]. A number of studies used this dataset to evaluate the performance of individual algorithms and hybrid algorithms [20]. Other studies opted to investigate a variety machine learning algorithms including Lasso, Ridge Regression, Mean value as a predictor, Random Forest (RF), SVR, ANN, GRU, and LSTM [31], [36], [39]. A summary of the experiments and results for this dataset can be found in Table 2-4.

Table 2-4: Experiment and results Summary for Rossmann's Sales Dataset

Paper	Algorithms	Results		
		MSE	**RMSE**	**RMSPE**
[20]	ARIMA		771	
	ARNN		565.58	
	SVR		666.7	
	XGBoost		670	
	ARIMA-ARNN		530.4	
	ARIMA-XGBoost		540.5	
	ARIMA-SVR		610	
	STL Decomposition		**426.4**	
[58]	SVR performed 29.41% better than the Softmax in terms of RMS.			
[36]	Mean Value	1.050		
	Lasso	0.286		
	Ridge Regression	0.200		
	RF	0.101		
	SVR	0.211		
	ANN	0.413		
	GRU	**0.072**		

Paper	Algorithms	Results		
	LSTM	**0.072**		
[39]	Linear Regression			52.7%
	SVR			12.8%
	RF			**12.3%**
[31]	Linear Regression			30.2%
	FDR			29.1%
	SVR			**17.4%**

2.5.5 Global Energy Prediction Competition - Wind Dataset

This dataset was used in the GEFCom2012 and depicts the energy production for seven windfarms as well as the wind data for each farm over the period 1 July 2009 to 31 December 2010 [59]. The wind data is recorded every 12 hours. The data demonstrates a stochastic and cyclic nature [32], [37]. Mangalova and Agafonov [37] applied k-nearest neighbour's (KNN) to the problem while Silva [32] carried out an experiment using a linear regression algorithm, K-Means clustering and Generalized Boosted Regression Algorithms at varying stages to obtain predictions. The table below summarizes the experiment results.

Table 2-5: Experiment and Results Summary for the GEFCom2012 - Wind Dataset

Paper	Algorithms	Results
		RMSE
[37]	KNN	0.1472
[32]	GBM	0.14567

2.5.6 Global Energy Predication Competition – Load Dataset

The GEFCom2012 Load dataset consists of hourly load and temperature data for an electric utility in the USA that has twenty zones, during the period 2004 to 2008 [59]. A parametric algorithm based on multiple linear regression algorithms, was tested on this dataset as well as a comparison between Kernelized Regression, Frequency Neural Network, deep FFNN, and deep RNN [33], [60]. A summary of the results can be seen in Table 2-6.

Table 2-6:Experiment and Results Summary for GEFCom2012- Load Dataset

Paper	Algorithms	Results		
		WRMSE	**RMSE**	**RMSPE**
[60]	Kernelized Regression		1540	8.3%
	Frequency Neural Network		1251	6.7%
	FFNN		1103	5.9%
	RNN		**530**	**2.8%**
[33]	Parametric Algorithm	67214		

2.5.7 Sterling Pound/United States Exchange Rate Dataset

This dataset illustrating the daily exchange rate as reported by the International Monetary Fund to the issuing central bank covers the period 1 January 1995 to 11 April 2018 [55]. The number of observations varied in the previous studies while Masum et al. [16] confirmed that the data is non-stationary. Some studies compared the ARIMA against the ANN and hybrid algorithms [16], [21], [30]. Table 2-7 captures a summary of the experiments and results for the Exchange Rate dataset.

Table 2-7: Experiment and Results Summary for Exchange Rate Dataset

Paper	Algorithms	Results	
		MSE	**RMSE**
[21]	ARIMA	4.52977 x 10-5	
	ARNN	4.52657 x 10-5	
	Proposed Algorithm	**4.35907 x 10-5**	
[30]	Compared with Zhang [21]		
	Proposed Algorithm	**3.76399 x 10-5**	
[16]	ARIMA		0.00561
[23]	Linear Regression, SVR, Decision Tree, Ada Boost, RF, KNN, Bagging regressor	Linear Regression performed the best for daily, monthly, yearly and weekly forecasts. KNN performed best for quarterly forecasts	

2.5.8 Stock Market Indices Dataset

This dataset includes the daily S&P500 indices over the period 4 January 2010 to 17 May 2019 [61]. This dataset involved experiments that compared the performance of single SVR and ANN algorithms to hybrid algorithms [62], [63]. A summary of these experiments and results are depicted in Table 2-8.

Table 2-8: Experiment and Results Summary for Stock Indices Dataset

Paper	**Algorithms**	**Results**			
		NMSE	**MAPE**	**MSE**	**rRMSE**
[63]	ANN		2.49	26636.56	3.10
	SVR-ANN		2.12	20455.22	2.72
	SVR		2.08	**18445.44**	**2.59**
	SVR-SVR		**2.06**	19013.75	2.62
	RF		2.40	24734.84	2.97
	RF-SVR		2.12	20067.88	2.69
[62]	SVR	0.9228			
	SVR Hybrid	**0.7817**			
[23]	Linear Regression, SVR, Decision Tree, Ada Boost, RF, KNN, Bagging regressor	Linear Regression performed the best for daily, monthly, yearly and weekly forecasts. KNN performed best for quarterly forecasts			

2.6 Key Findings

The machine learning pipeline that majority of the studies followed incorporated the loading and exploration of data followed by the data preprocessing. Thereafter, the data was split, and a model was trained using the training data. Validation data was used to tune the model hyperparameters to obtain the optimal model. Once the model training performance was acceptable and the optimal parameters were realised, the model was fitted using the entire training data set and predictions were performed using the test data. Once this was complete, the model performance was evaluated. The reviewed literature compared the performance of various machine learning algorithms, while the robustness and stability of the algorithms were not discussed.

The experimental process for studies evaluating a single proposed algorithm against existing research followed the machine learning process mentioned in Section 2.3.1, for each dataset. Most authors

opted to carry out different preprocessing for each dataset based on the algorithm being evaluated, while Samsudin et al. [27] carried out the same preprocessing for each dataset irrespective of the algorithm being evaluated.

A variation of evaluation techniques was adopted in the studies reviewed. These were dependent on the datasets and research benchmarks used by the authors. The predominant evaluation metrics in the reviewed studies were the MSE and RMSE. A common challenge encountered with comparing results from the different studies was the use of different evaluation metrics across the studies. In the case where the same metrics are used, the scale of the evaluation metrics is not compatible, possibly due to different preprocessing techniques being used during the data transformation and scaling process. It must be noted that RMSE and MSE are unit specific and therefore cannot be used to make comparisons across datasets. The RMSPE and MAPE are better suited to comparing algorithm performance on different datasets as they are percentage-based. However, these metrics can be unreliable when the actual data is zero or close to zero. The MASE was found to overcome the shortcomings of the MAPE and RMSPE.

The training and test partitioning of data in the reviewed literature was application-specific especially for the smaller datasets. The Canadian Lynx dataset was prevalently split using an 88:12 percent partitioning ratio while the Sunspot data was split using a 77:23 ratio. The Airline dataset was predominantly split using a 92:8 percent ratio. Common partitioning ratios that were used in studies is a 70:30 percent ratio and 75:25 percent ratio. It was noted that most studies used a portion of the training data for validation and hyperparameter tuning.

Adhikari and Agrwal [5] and Samsudin et al. [27] carried out research comparing various time series prediction algorithms on several datasets with different characteristics, while Zhang [21], Ghiassi et al. [28], Khashei and Bijari [30] and Aladag et al. [22] aimed to propose a hybrid algorithm that performed better than algorithms at its simplest form. The studies concerning the Rossmann's Sales, Load and Wind datasets focused on solving the prediction problem for a specific application. The most predominant machine learning algorithms used in studies is the ANN and SVR. Comparison of the algorithms performance for basic implementations indicates the ANN performs the best for the Airline, Sunspot, Stock Indices, and Exchange Rate datasets. The Rossmann's Sales and Canadian Lynx datasets showed promising performance for the SVR.

The LSTM is found to be the most suitable for time series data by Kraus et al. [36], Borovykh et al. [42] and He [43]. The studies using the datasets in this review lack the comparison between the LSTM and other algorithms. The reviewed literature also lacks an analysis of the robustness and stability of the

algorithms. A summary of the characteristics of the various datasets is depicted in Table 2-10 while Table 2-9 shows the rank of the various machine learning algorithms according to the application.

Table 2-9: Rank of Algorithms According to Dataset

	Airline	**Sunspot**	**Canadian Lynx**	**Rossmann's Sales**	**GEFCom 2012 Load**	**GEFCom 2012 Wind**	**Exchange Rate**	**Stock Indices**
1	ANN	ANN-ARIMA [30]	LSSVR	Hybrids	Parametric Algorithm	GBM	ANN-ARIMA [30]	ANN-SVR
2	SVR	DAN2	SVR	Random Forest	RNN	KNN	ANN-ARIMA [21]	SVR-RF
3	DAN2	ANN-ARIMA [21]	ERNN-ARIMA [22]	SVR	FFNN		ANN	SVR-SVR
4	SARIMA	ANN	ANN	ANN	FNN		Linear Regression	ANN
5	SANN	SVR	ANN-ARIMA [21]		Kernelized Regression		SVR	SVR

Table 2-10: Dataset Characteristics

Dataset	**Characteristics**
Airline	Non-stationary, multiplicative seasonal pattern with an upward trend, non-linear, small number of observations
Sunspot	Non-stationary, non-linear, non-gaussian, cyclic pattern with a mean of 11 years, a small number of observations
Canadian Lynx	Non-stationary, a small number of observations
Rossmann's Sales	Non-linear, weekly seasonality, cyclic trend, medium number of observations
GEFCom2012 Load	Medium number of observations
GEFCom2012 Wind	Stochastic, cyclic, medium number of observations
Exchange Rate	Medium number of observations, non-stationary
Stock Indices	A small number of observations

Chapter 3

3 Experimental Design

This chapter presents the methodology and design of the experiments conducted in this research. The chapter describes the experimental workflow, as well as each step of the workflow in detail.

3.1 Experimental Process

Two experiments were carried out for each of the datasets, namely:

1. Identification of parameter values for the optimal model and evaluating the performance and stability of the model.
2. Evaluation of the robustness of the optimal model.

Figure 3-1 presents the high-level experimental approach taken in this study. This workflow is carried out for each of the datasets and is based on the process adopted by reviewed literature, as seen in Figure 2-1.The workflow was altered for this study to cater for Experiment 2.

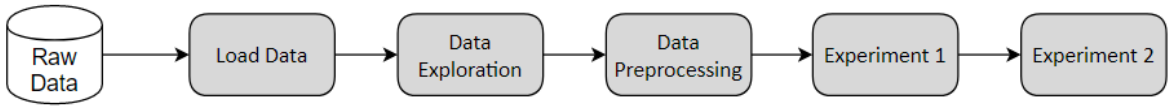

Figure 3-1: Experiment Workflow Diagram

Experiment 2 followed Experiment 1, as the optimal hyperparameter values identified in Experiment 1 were used for Experiment 2.

3.2 Data Exploration

Once the raw data was loaded, the data was examined to gather more information on the dataset such as the size, number of features and the type of data. The exploration of the data aids in better understanding the problem. The data was visualised and investigated for any characteristics or limitations that could affect the algorithm performance. This includes characteristics such as stationarity, linearity and trends. The seasonality, linearity and trends were identified by decomposing the time series into its seasonality, trends and noise components. The stationarity of the time series data was tested using a unit root test.

3.3 Data Preprocessing

A general feature engineering approach was applied across all the datasets, while the preprocessing methods that were applied specifically to each dataset was based on the methods used in the reviewed literature.

Feature engineering was carried out by extracting statistical features from the time series. Using lag features and windowing of statistical features, allows the time series to be transformed into a supervised learning problem. For the experiments in this study, ten lags for the response variable were used in order to capture any trends and patterns. The window features are features that occur over a fixed window of previous time steps. This was done by using a rolling window and expanding window of three, in order to obtain the minimum, maximum, mean and standard deviation.

The adopted approach involved using the same preprocessing rules on a dataset for all algorithms evaluated on that dataset, similar to what was done by Samsudin et al. [27]. The Canadian Lynx dataset was transformed using a logarithm to the base 10, following Khashei and Bijari [30] and Zhang [21].

The Airline Passenger dataset was normalised between [0, 1] using the min-max normalisation. However, the data was transformed back to its original scale after prediction. This was the same approach taken by Samsudin et al.[27].

The Sunspot dataset was not scaled as this was the approached followed by Ghiassi et al. [21], Khashei and Bijari [22] and Zhang's [15]. This produced poor results that were not in the same range as previous studies. The data was then normalised using the min-max normalisation and transformed back to its original scale for evaluation.

A natural logarithm was used to transform the Exchange rate dataset, as this was the approach followed by Khashei and Bijari [30] and Zhang [21]. The data was resampled as weekly observations rather than daily as this was the format of the data used in research by Zhang [21], Dingli and Fournier [23] and Khashei and Bijari [30]. Since the markets close on weekends and the exchange rate and stock prices remain the same as the Friday, the data was backfilled for Saturdays and Sundays as well as public holidays.

The Load dataset and Stock Indices dataset were normalised and then converted back to the original scale after prediction. This approach follows what was done in the research by Dingli and Fournier [23]. For the Load dataset, the load consumption data and temperature data were combined into a single table. Only one wind farm from the Wind Power dataset was used for this experiment.

A sample of a single store was used from the Rossman's Sales dataset for this experiment in order to reduce the dataset size based on the available processing power. The data was normalised and then converted back to the original scale after prediction.

3.4 Evaluation Metrics

The MSE, RMSE and MAE have been selected for this research as it is prominently used for the various datasets in reviewed literature. These evaluation metrics that are defined in Section 2.3.5, were used to compare the results in this study to the results of the benchmarked studies. The MASE was selected to compare the performance of the algorithms across all datasets due to its advantages described in Section 2.3.5.

3.5 Experiment 1

3.5.1 Overview

Experiment 1 focuses on evaluating the model performance and stability as well as finding the optimal model hyperparameter values. Figure 3-2 shows the machine learning process for Experiment 1 which was derived from the workflow seen in Figure 2-1.

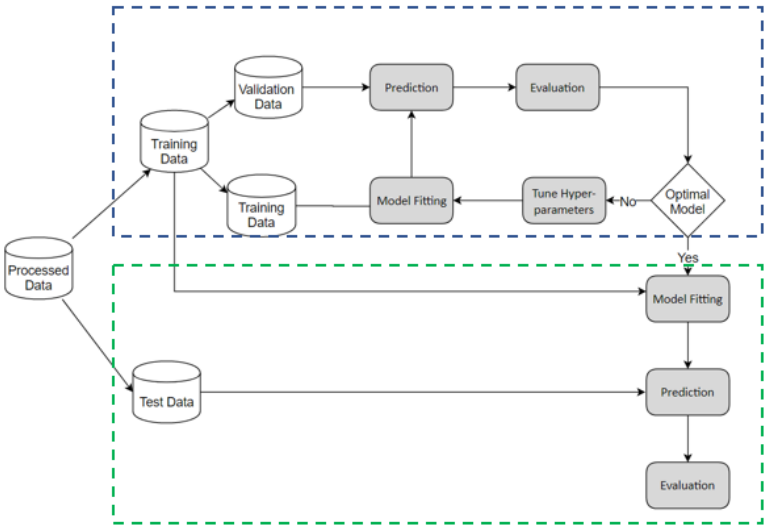

Figure 3-2: Experiment 1 Workflow Diagram

The processed data was split into training and test datasets which were normalised. A portion of the training dataset was used as a validation dataset, to carry out hyperparameter tuning. Once the optimal model was found, the model was fitted using the entire training set and it was evaluated using the unseen test dataset. Algorithms such as the ANN and LSTM have a stochastic nature, consequently enabling them to model non-linear datasets more accurately. This characteristic impacts the stability of the model and depending on the seed that is selected, the model can yield varying results. The

stability of the model refers to how much the accuracy deviates from the mean when the same model is run multiple times. In order to effectively evaluate the performance and stability of the stochastic machine learning algorithms, Experiment 1 was repeated for ten runs for the ANN and LSTM, and the mean and the deviation of the results were recorded. If the deviation from the mean is large, this indicates that the model is unstable.

3.5.2 Dataset Partitioning

The dataset was split into a training and test set. The test set in this experiment was unseen data at the end of the dataset, that was used to evaluate the model. The training set was further split into training and validation sets, which was used for hyperparameter tuning. Once the optimal model parameters were identified the model was then trained using the entire training set and evaluated using the unseen test data.

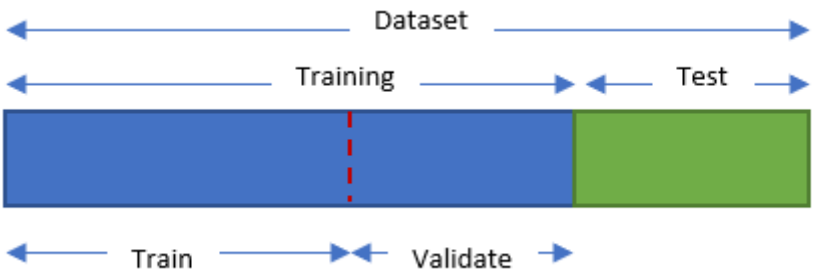

Figure 3-3: Overview of Dataset Partitioning

3.5.2.1 Training and Test

Each dataset was split into a training and test set. This was done using the hold out method according to the ratios used in the reviewed literature, as described in Section 2.3.3.1.

88 percent of the Canadian Lynx dataset were used for training while 12 percent was used for testing. This approach was used by Zhang [21]. The training set for the Airline Passenger data was 92 percent of the dataset while the test set was 8 percent, which was the approach Adhikari and Agrawal [6], Ghiassi et al. [7] and Khashei and Bijari [2] used. The Sunspots dataset was split using a 77:23 percent ratio which was a similar to the approach taken by Adhikari and Agrawal [6], Ghiassi et al. [7], Khashei and Bijari [2] and Zhang [3]. 12 months of the Exchange Rate dataset was used for testing. This was the approach used by Khashei and Bijari [2] and Zhang [3]. Two years of data for the Rossmann's Sales dataset was used for training and 7 months for testing. This was done to capture the yearly seasonality for 2 years. The Load dataset and Wind dataset were split according to the 75:25 percent training test ratio. The Stock Indices dataset was split such that 5 months of data was used for testing.

The table below summarises the dataset partitioning into training and test, for each dataset.

Table 3-1: Dataset Partitioning Summary for Training and Test Datasets

Dataset Name	Total No. of Elements	No. of Elements in Training Set	Training (%)	No. of Elements in Test Sets	Test (%)
Airline	144	132	92	12	8
Sunspots	288	221	77	67	23
Canadian Lynx	114	100	88	14	12
Rossman's Sales	942	730	78	212	22
Wind	13176	9882	75	3294	25
Load	8784	6588	75	2196	25
Exchange Rate	1218	1166	96	52	4
Stock	3420	3284	96	136	4

3.5.2.2 Validation Data for Hyperparameter Tuning

The training dataset for each dataset was split further into a training dataset and validation dataset which was used for hyperparameter tuning. For the ANN and LSTM, the hold out method was used for hyperparameter tuning, where 25 percent of the training data was used for validation. The prequential approach discussed by Cerqueira et al [34] was used on the training dataset for hyperparameter tuning of the SVR. This method splits the data into blocks and only uses observations that occur before the given validation data observations, allowing the data to be kept in chronological order. The following figure depicts how this method is applied for 5 training/test iterations with 6 blocks of data. For the first training/test iteration, only the first two blocks are used. The first block is used for training and the second block is used for validation. For the second training/test iteration, 3 blocks are used. The first two for training and the third for validation. This process continues until all the blocks are used.

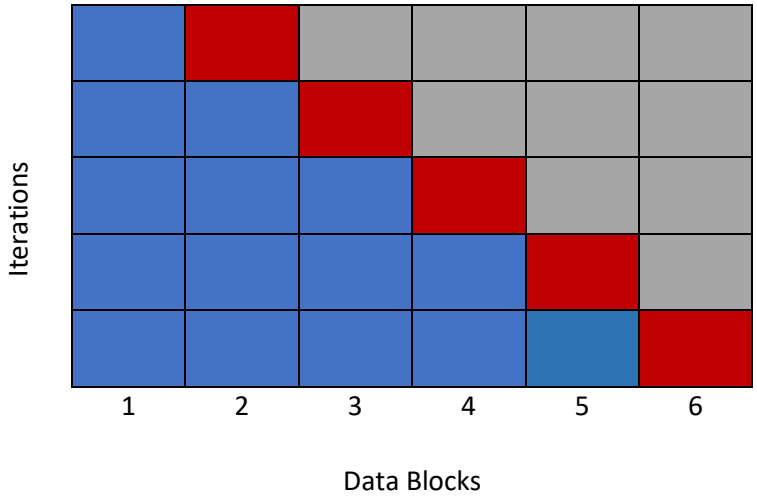

Figure 3-4: Prequential dataset partitioning method

The blue blocks indicate the training data, the red blocks are the validation data while the grey blocks show the data that is not used in the training/test iteration. The figure above depicts the data blocks for 5 training/test iterations.

3.5.3 Hyperparameter Tuning

Hyperparameter tuning for the SVR was carried out using ten train/test iterations. This means that the data was broken up into 11 blocks of data as described in Section 3.5.2.2. The hyperparameter value ranges that were selected for the SVR are seen in the table below. The SVR hyperparameter values used by Samsudin et al. [27], was used as an initial starting point and adjusted through trial and error to find hyperparameter values that produced acceptable results.

Table 3-2: SVM Parameter Ranges

Parameter	Range	Increment
C	[1, 15]	1
ε	[0.001, 0.01]	0.01
γ	[0.0001, 0.1]	0.0001

The radial basis function kernel (RBF) was selected, in accordance with [27], [5], [38] where the RBF was shown to have the best performance. As described in Section 2.3.4, the RBF has the advantage that it can be used for both linear and non-linear modelling. It also requires low computational power for hyperparameter tuning in comparison to other kernels.

The hyperparameters for the ANN and LSTM, such as the number of hidden neurons and the number of layers were determined manually through experimentation. This followed the approach used by Samsudin et al [27].

The hyperparameter value ranges tested for the ANN is shown in the table below.

Table 3-3: Summary of ANN Hyperparameters

Parameter	Range	Increment
Number of Layers	[1, 6]	1
Number of Nodes	[10, 50]	10
Epochs	1200	-

Three different activation functions were also explored. In comparison to the sigmoid and linear activation functions, the relu activation function yielded best results for the ANN.

The hyperparameter value ranges for the LSTM can be seen in the table below.

Table 3-4: Summary of LSTM Hyperparameters

Parameter	Range	Increment
Number of Layers	[1, 3]	1
Number of Nodes	[20, 500]	10
Epochs	1200	-

For the LSTM the hyperbolic tangent activation function, which is the default activation function in Keras, was used.

Too many epochs can lead to overfitting and too few epochs can lead to underfitting. To address this issue an early stopping monitor of 200 epochs was used for both the LSTM and the ANN. The best model from all the epochs was selected rather than the model from the last epoch.

3.6 Experiment 2

3.6.1 Overview

Once the optimal hyperparameter values were identified in Experiment 1, they were used in Experiment 2 along with the preprocessed data to interrogate the robustness of the models. It is noted that the test is biased as the test is carried out on seen data. However, the objective of the test is to determine the resilience of the model. The model robustness refers to how much the performance of a model differs when it is trained and tested on slightly different data. Slightly different data can

include a different dataset with the same features, the same dataset with noise or the same dataset over different time periods. For this experiment the same dataset was used for different time periods. In order to partition the data, the experiment was repeated for ten training/test iterations in order to carry out testing over ten different periods in the data. The figure below shows the process for Experiment 2.

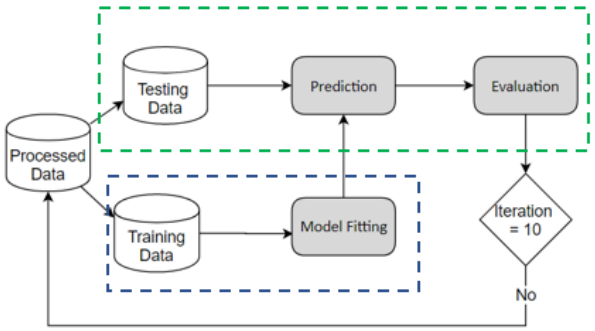

Figure 3-5: Experiment 2 Workflow Diagram

3.6.2 Dataset Partitioning

The entire dataset was partitioned using the prequential method described in Section 3.5.2.2. The prequential method was used for ten training/test iterations and partitioned the data into a training and unseen test set. Therefore, there were ten different dataset partitions tested. According to Figure 3-4, this means that the data was broken up into 11 blocks of data and for the final iteration ten blocks of data was used for training and one for testing.

3.6.3 Hyperparameter Tuning

There was no hyperparameter tuning carried out for Experiment 2. The optimal model parameter values identified through hyperparameter tuning in Experiment 1, were used for Experiment 2.

3.7 Summary

There were two experiments carried out for each model for the various datasets. The first experiment aims to evaluate the model performance and stability as well as finding the optimal model parameters. The entire workflow for Experiment 1 that is seen in Figure 3-2 was repeated for ten runs for the ANN and LSTM due to the instability of neural networks. The second experiment focuses on evaluating the robustness of the models. Preprocessing differed for each dataset based on existing studies. However,

feature engineering was performed to obtain lag features and window features. The MSE, RMSE, MAE and the MASE were selected as evaluation metrics for both the experiments.

For Experiment 1 each dataset was split into a training set and test set using the hold out method. The ratios were selected according to existing literature. For datasets that differed from literature or if the ratio was not specified, a 75:15 percent ratio was used. The training set was split further into a training and validation set which was used for hyperparameter tuning. A prequential method was used for the SVR while the hold out method was used for the ANN and LSTM. The performance of the models was measured using the selected evaluation metrics while the stability was measured using the deviation of the evaluation metrics from the mean.

For Experiment 2 the dataset was split using the prequential method such that there were ten different dataset partitions. The optimal model parameters identified in Experiment 1 were used for Experiment 2. The robustness of the models was evaluated with the mean and standard deviation of the selected evaluation metrics.

Chapter 4

4 Results

The results obtained from the experiments described in Chapter 3, is presented in the following chapter. For Experiment 1, a description of the dataset characteristics that were identified in the exploratory data analysis are presented for each dataset as well as a plot of the actual values vs predicted values from the best performing models. There are two results tables for each dataset. The first table presents the evaluation metric values. This includes the RMSE for the training set; and the MSE, RMSE, MAE and MASE for the test set. Experiment 1 was repeated for 10 iterations for the ANN and LSTM, and the mean and the deviation of the results were recorded. The improvement over the best performing benchmarked results is presented in the second table. For experiment 2 there are three results tables for each of the algorithms. The chapter is concluded with three summary tables. The first table summarises the dataset characteristics, the second summarises the MASE results and the third summarises the model hyperparameters.

4.1 Airline Passenger Dataset

4.1.1 Experiment 1

From the exploratory data analysis, it was found that the Airline Passenger dataset is a small, nonlinear and univariate dataset. The dataset demonstrates an upward trend, as seen in Figure 4-1. The unit root test showed that the data is non-stationary. A multiplicative yearly seasonality was observed for this dataset.

Using the hyperparameter value ranges and tuning approaches specified in Section 3.5.3, the optimal parameters found for the SVR were: $C = 13$, $\varepsilon = 0.009$, $\gamma = 0.1$. The optimal architecture for the LSTM was a single layer with 50 nodes while the ANN had 5 layers with 25, 60, 120, 50 and 20 nodes respectively.

Figure 4-1 shows the actual airline passengers as well as the number of passengers predicted by the best performing model on the test dataset. The experiment results for the Airline Passenger dataset are presented in Table 4-1. A lower value for the evaluation metrics indicates a higher algorithm accuracy. The LSTM outperformed the ANN and SVR. For the experiments with multiple iterations, a minimal deviation from the mean indicates the model is stable while a high deviation indicates poor stability. The ANN performed the worst and had a higher deviation in the results when compared to

the LSTM. The range of the results for the LSTM over ten runs is minimal, indicating that the model is more stable than the ANN.

The improvement on the benchmarked results is presented in Table 4-2. The improvement was measured for the RMSE as this was a common metric among all the benchmarked datasets. The table presents the actual improvement in the RMSE as well as the percentage improvement for the RMSE. The LSTM in this study, demonstrated an improvement of 48.20% and 46.41%, in comparison to the RMSE of the SVR and SARIMA in the Adhikari and Agrawal [5] study.

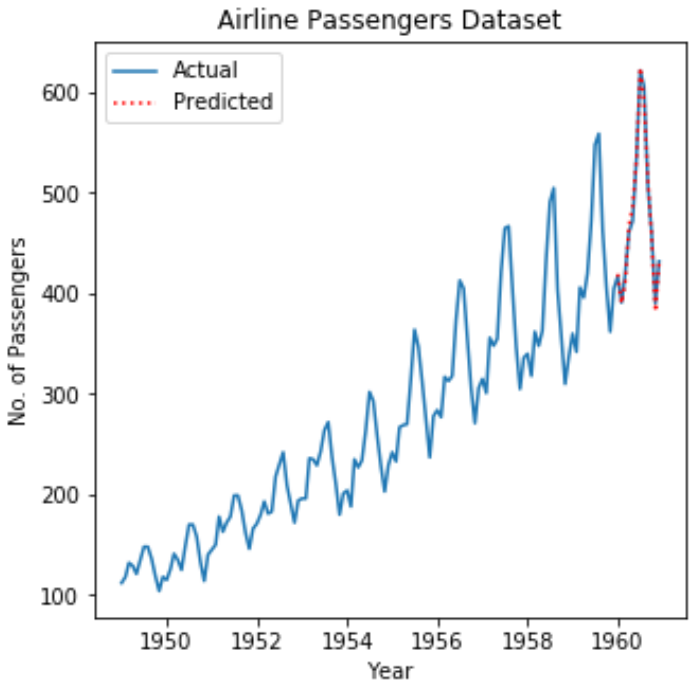

Figure 4-1: Airline Passengers Dataset

Table 4-1: Airline Passenger Dataset Results

	Train	**Test**			
	RMSE	**MSE**	**RMSE**	**MAE**	**MASE**
SVR	5.02	130.42	11.42	9.58	0.364
ANN	8.89 ± 6	304.59 ± 437	16.45 ± 11	14.02 ± 10	0.533 ± 0.4
LSTM	**3.84 ± 0.32**	**51.09 ± 11**	**7.13 ± 0.76**	**5.68 ± 1.12**	**0.22 ± 0.04**

Table 4-2: Improvement on Best Benchmarked Results for Airline Dataset

	Literature RMSE	**Actual RMSE**	**Δ RMSE**	**% Improvement**
SARIMA [5]	13.759865	7.127524	6.632341	48.20
SVR [5]	13.299823	7.127524	6.172299	46.41

4.1.2 Experiment 2

The results obtained from Experiment 2, for the three optimal models can be seen in the tables below. It is noted that the results are not always in the range of the results from Experiment 1, for the initial iterations as only a small portion of the dataset is used for training. The latter iterations show values that are within the same range as the results presented in Experiment 1. This depicts that the models are robust when presented with approximately the same split of training and testing data. Table 4-6 summarises the results from Experiment 2 for the different algorithms. The table presents the mean and standard deviation of the last 5 iterations for each of the evaluation metrics. From Table 4-6 it can be seen that the standard deviation for the SVR is much lower than that of the LSTM and ANN, indicating that the SVR is more robust over a different range of training/testing splits.

Table 4-3: SVR Experiment 2 Results for Airline Dataset

Split	MSE	RMSE	MAE	MASE
0	1616.833603	40.209869	35.490895	0.546014
1	632.554500	25.150636	22.196736	1.021322
2	486.037446	22.046257	17.769078	1.017452
3	248.497357	15.763799	12.320940	0.731054
4	679.645441	26.070010	17.149351	0.966665
5	252.149279	15.879209	10.036563	0.527826
6	116.206085	10.779893	9.766193	0.486183
7	91.487197	9.564894	6.842339	0.311320
8	296.925640	17.231530	12.007882	0.497007
9	196.672266	14.023989	11.364974	0.434586

Table 4-4: ANN Experiment 2 Results for Airline Dataset

Split	MSE	RMSE	MAE	MASE
0	324.225098	18.006252	15.636113	0.240556
1	66.313476	8.143309	7.332211	0.337372
2	209.010645	14.457200	12.464209	0.713697
3	147.004618	12.124546	9.856691	0.584840
4	162.258910	12.738089	10.266634	0.578704
5	241.864576	15.551996	10.971136	0.576975
6	485.981808	22.044995	17.580446	0.875193
7	763.692375	27.634985	22.100682	1.005559
8	274.788736	16.576753	13.627178	0.564030
9	217.910977	14.761808	13.577252	0.519182

Table 4-5: LSTM Experiment 2 Results for Airline Dataset

Split	MSE	RMSE	MAE	MASE
0	925.195910	30.417033	26.801910	0.412337
1	274.058760	16.554720	14.705020	0.676611
2	270.354467	16.442459	14.908490	0.853656

Split	MSE	RMSE	MAE	MASE
3	327.390412	18.093933	14.034236	0.832712
4	70.670710	8.406587	6.552851	0.369367
5	383.508833	19.583382	17.776780	0.934886
6	502.295142	22.411942	18.078177	0.899971
7	84.419215	9.187993	8.509343	0.387167
8	68.606906	8.282929	6.911189	0.286055
9	39.036558	6.247924	5.242880	0.200483

Table 4-6: Summary of Experiment 2 Results for Airline Dataset (Last 5 iterations)

	MSE		RMSE		MAE		MASE	
	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev
SVR	**190.69**	**87.30**	**13.50**	**3.27**	**10.00**	**2.00**	**0.45**	**0.085**
ANN	396.85	231.01	19.31	5.46	15.57	4.35	0.71	0.218
LSTM	212.35	212.35	13.14	7.32	11.30	6.16	0.54	0.350

4.2 Sunspots Dataset

4.2.1 Experiment 1

From the exploratory data analysis, it was observed that the Sunspots dataset is a small dataset that shows non-linearity. The dataset is univariate and demonstrates a cyclic trend. The unit root test showed that the data is non-stationary. There was no seasonality observed. Figure 4-2 shows the actual and predicted values for the Sunspot dataset. It is observed that the prediction for the last year is much lower than the actual value.

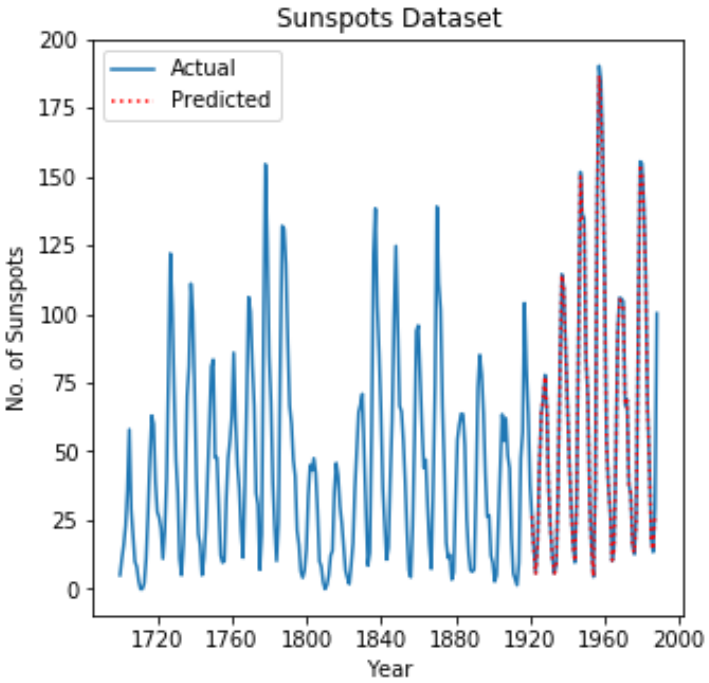

Figure 4-2: Sunspots Dataset

The optimal hyperparameter values found for the SVR model was C = 2, ε = 0.002, γ = 0.1. The optimal architecture for the LSTM was a single layer with 50 nodes, while the ANN had three layers with 10, 30 and 20 nodes respectively.

The performance results for the Sunspots dataset are presented in Table 4-7. The SVR and the ANN produced similar results. However, when considering the ANN's worst case scenario, the SVR is superior. The LSTM demonstrated the best performance, with an RMSE of 1.83. The LSTM also had a better stability than the ANN, as depicted by the range of results over ten runs.

The improvement on the benchmarked results is presented in Table 4-8. The greatest improvement of 90.20% was seen for the ANN in Adhikari and Agrawal's [5] study.

Table 4-7: Sunspots Dataset Results

	Train	**Test**			
	RMSE	**MSE**	**RMSE**	**MAE**	**MASE**
SVR	3.87	93.95	9.69	4.80	0.30
ANN	3.42 ± 2.38	49.76 ± 43	6.82 ± 2.84	4.80 ± 1.88	0.30 ± 0.12
LSTM	**1.83 ± 0.26**	**3.30 ± 1.45**	**1.79 ± 0.39**	**1.13 ± 0.31**	**0.07 ±0.02**

Table 4-8: Percentage Improvement on Best Benchmarked Results for Sunspot Dataset

	Literature RMSE	**Actual RMSE**	**Δ RMSE**	**% Improvement**
ANN [5]	18.280400	1.791564	16.489	90.20
DAN2 [28]	16,062378	1.791564	14.271	88.85
ARIMA-ANN [30]	15,303793	1.791564	13.512	88.30
ARIMA-ANN [21]	16,737967	1.791564	14.946	89.29

4.2.2 Experiment 2

The results obtained from Experiment 2 for the three optimal models can be seen in the tables below. It is observed that the results for the last iteration for the SVR and ANN differs greatly from the results presented in Experiment 1. In this scenario, the LSTM is the most robust model as the results for the latter iterations are similar to Experiment 1 and the standard deviation seen in Table 4-12 is minimal.

Table 4-9: SVR Experiment 2 Results for Sunspots Dataset

Split	**MSE**	**RMSE**	**MAE**	**MASE**
0	508.998066	22.560985	19.625039	1.543542
1	192.767192	13.884063	8.380015	0.554696
2	129.057305	11.360339	7.804417	0.452430
3	10.657922	3.264647	2.666296	0.159595
4	23.088460	4.805045	4.032999	0.266005
5	10.713499	3.273148	2.350983	0.144953
6	2.673230	1.635002	1.149529	0.069850

Split	MSE	RMSE	MAE	MASE
7	3.776163	1.943235	1.350422	0.084161
8	178.941949	13.376919	7.161755	0.444140
9	6.773248	2.602546	2.019429	0.115706

Table 4-10: ANN Experiment 2 Results for Sunspots Dataset

Split	MSE	RMSE	MAE	MASE
0	488.794018	22.108686	18.490961	1.454345
1	237.328525	15.405471	10.791346	0.714308
2	400.343438	20.008584	16.672647	0.966530
3	227.324082	15.077270	13.170218	0.788324
4	224.138445	14.971254	12.098576	0.797987
5	26.312183	5.129540	3.786140	0.233439
6	17.585409	4.193496	3.399555	0.206571
7	4.408534	2.099651	1.612693	0.100506
8	118.781544	10.898695	6.926095	0.429525
9	172.740703	13.143086	11.030621	0.632017

Table 4-11: LSTM Experiment 2 Results for Sunspots Dataset

Split	MSE	RMSE	MAE	MASE
0	241.280236	15.533198	12.665563	0.996168
1	32.751157	5.722863	4.415695	0.292287
2	42.693241	6.534006	4.918269	0.285117
3	2.032881	1.425791	1.252113	0.074947
4	13.579396	3.685023	2.661692	0.175558
5	1.497032	1.223533	1.001790	0.061767
6	0.532921	0.730014	0.545671	0.033157
7	0.800378	0.894639	0.648278	0.040402
8	4.760407	2.181836	1.574254	0.097628
9	1.112083	1.054554	0.863653	0.049484

Table 4-12: Summary of Experiment 2 Results for Sunspots Dataset (Last 5 iterations)

	MSE		RMSE		MAE		MASE	
	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev
SVR	40.58	77.41	4.57	4.97	2.81	2.48	0.17	0.155
ANN	67.97	79.95	7.09	4.70	5.35	3.71	0.32	0.211
LSTM	**1.74**	**1.73**	**1.22**	**0.53**	**0.93**	**0.40**	**0.06**	**0.025**

4.3 Canadian Lynx Dataset

4.3.1 Experiment 1

The Canadian Lynx dataset is a small, univariate dataset that shows a non-linear shape. The unit root test confirmed that the dataset is non-stationary. There is a cyclic trend visible in the dataset. Figure 4-3 shows the actual number of Canadian Lynx and the values predicted by the best performing model.

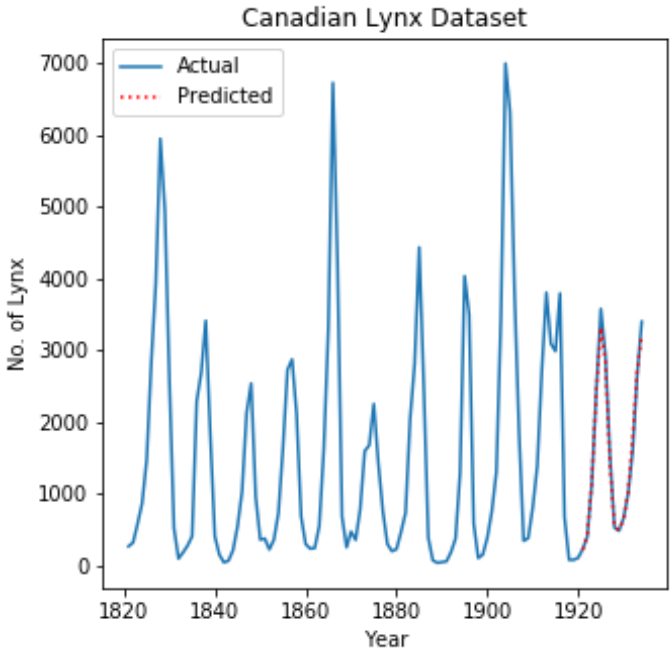

Figure 4-3: Canadian Lynx Dataset

The optimal hyperparameter values found for the SVR model was C = 13, ε = 0.008, γ = 0.1. The optimal architecture found for the LSTM was a single layer with 50 nodes while for the ANN, four layers with 10, 50, 100 and 20 nodes were found to be optimal.

The performance results for the Canadian Lynx dataset are presented in Table 4-13. The results are presented according to the logarithm to the base 10, in order to be comparable to the benchmarked literature. The SVR performed the best during training. However, it was outperformed by the LSTM, by a small margin, for the out of sample data. It is noted that for the worst-case scenario, over the ten runs that the LSTM was tested for, the SVR performs better than the LSTM. It was also noted that although the LSTM's MASE value, was slightly higher than that of the SVR, in terms of the remaining evaluation metrics the LSTM performed better. The ANN performed the worst for this dataset. The improvement on the benchmarked results is presented in Table 4-14. The LSTM in this study showed an improvement of 59.38% on the LSSVR [38], which was the best performing model in the reviewed literature.

Table 4-13: Canadian Lynx Dataset Results

	Train	Test			
	RMSE	**MSE**	**RMSE**	**MAE**	**MASE**
SVR	0.050	0.001	0.024	0.018	0.061
ANN	0.170 ± 0.09	0.017 ± 0.019	0.127 ± 0.063	0.101 ± 0.043	0.334 ± 0.143
LSTM	**0.058 ± 0.006**	**0.001 ± 0.0002**	**0.022 ± 0.005**	**0.018 ± 0.003**	**0.061 ± 0.009**

Table 4-14: Percentage Improvement on Best Benchmarked Results for Canadian Lynx Dataset

	Literature RMSE	Actual RMSE	Δ RMSE	% Improvement
AR [5]	0.071577	0.022249	0.049328	68.92
ARIMA-RNN [22]	0,094868	0.022249	0.072619	76.55
DAN2 [28]	0,077459	0.022249	0.05521	71.28
ARIMA-ANN [30]	0,077459	0.022249	0.05521	71.28
ARIMA-ANN [21]	0,131274	0.022249	0.109025	83.05
SETAR [57]	0.122	0.022249	0.099751	81.76
LSSVR [38]	0,054772	0.022249	0.032523	59.38

4.3.2 Experiment 2

The results from Experiment 2 for the three optimal models can be seen in the tables below. It is noted that the results are not always in the range of the results presented above for the initial iterations, as only a portion of the dataset is used. The latter iterations show values that are within the same range as the results presented above which indicates that all the models are robust when applied to a training dataset of a similar size as used in Experiment 1. However, the means of the last 5 iterations are slightly higher than the results from Experiment 1. From Table 4-18, it can be seen that the SVR and LSTM are more robust than the ANN. Although the standard deviations for the SVR and LSTM are similar, the standard deviation of the MASE is lower for the SVR, indicating that the SVR is slightly more robust than the LSTM.

Table 4-15: SVR Experiment 2 Results for Canadian Lynx

Split	MSE	RMSE	MAE	MASE
0	0.167659	0.409462	0.348207	0.194453
1	0.071601	0.267584	0.212183	0.352512
2	0.002551	0.050512	0.042527	0.094502
3	0.014202	0.119171	0.090965	0.234431
4	0.003479	0.058986	0.049610	0.134301
5	0.008661	0.093063	0.072520	0.211412
6	0.013184	0.114822	0.089460	0.256657
7	0.008868	0.094168	0.077321	0.218212
8	0.001310	0.036201	0.033763	0.099051
9	0.000219	0.014804	0.012080	0.035587

Table 4-16: ANN Experiment 2 Results for Canadian Lynx

Split	MSE	RMSE	MAE	MASE
0	0.271127	0.520699	0.416400	1.511471
1	0.108449	0.329316	0.288634	0.919086
2	0.023068	0.151881	0.124705	0.415678
3	0.042837	0.206971	0.157079	0.543973
4	0.046401	0.215410	0.192374	0.657181
5	0.137978	0.371454	0.327866	1.157989
6	0.212382	0.460850	0.424356	1.433454
7	0.066291	0.257471	0.232028	0.754861
8	0.052421	0.228957	0.189912	0.631888
9	0.016092	0.126855	0.106757	0.352117

Table 4-17: LSTM Experiment 2 Results for Canadian Lynx

Split	MSE	RMSE	MAE	MASE
0	0.343326	0.585940	0.499160	1.811877
1	0.023086	0.151941	0.123834	0.394319
2	0.076595	0.276758	0.248376	0.827907
3	0.017476	0.132197	0.107425	0.372020
4	0.004038	0.063544	0.050438	0.172304
5	0.017671	0.132931	0.116963	0.413102
6	0.023462	0.153174	0.133280	0.450215
7	0.004007	0.063304	0.047694	0.155162
8	0.003878	0.062270	0.054396	0.180989
9	0.000499	0.022339	0.017528	0.057814

Table 4-18: Summary of Experiment 2 Results for Canadian Lynx Dataset (Last 5 iterations)

	MSE		RMSE		MAE		MASE	
	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev
SVR	**0.01**	**0.01**	**0.07**	**0.04**	**0.06**	**0.03**	**0.16**	**0.093**
ANN	0.1	0.08	0.29	0.13	0.26	0.12	0.87	0.430
LSTM	0.01	0.01	0.09	0.05	0.07	0.05	0.25	0.171

4.4 Rossmann's Sales Dataset

4.4.1 Experiment 1

The Rossmann's Sales dataset is a non-linear dataset that is medium size in comparison to the other datasets in this study. The seasonality was tested for several individual stores and a weekly seasonality was demonstrated. The Augmented Dickey Fuller unit root test showed that the dataset was stationary. However, since the dataset has seasonality, it is non-stationary. The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test confirmed that the dataset is non-stationary. The actual values and predicted values from the best performing model can be visualised in Figure 4-4.

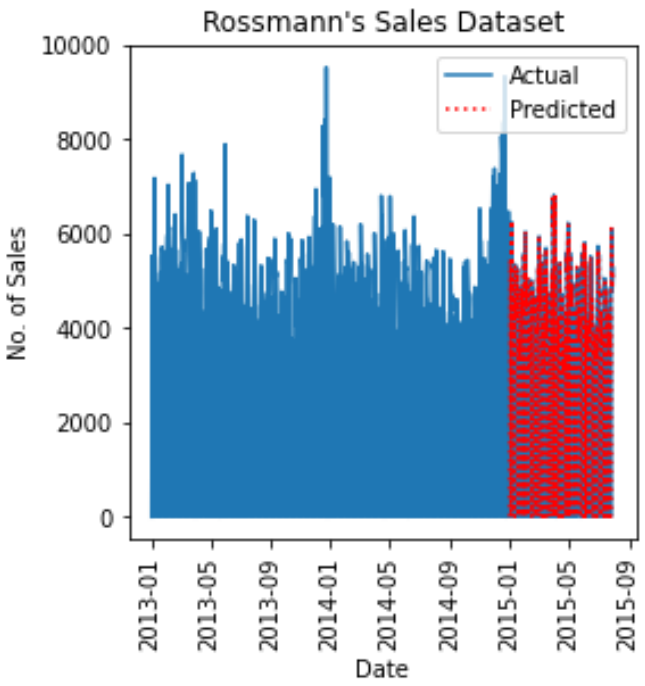

Figure 4-4: Rossmann's Sales Dataset - Store 1

The optimal hyperparameter values found for the SVR model was C = 3, ε = 0.02, γ = 0.006. The optimal architecture found for the LSTM was a single layer with 50 nodes while for the ANN, three layers with 30, 100 and 20 nodes were found to be optimal.

The experiment results for the Rossmann's Store dataset are presented in Table 4-19. The LSTM performed the best for the dataset while the SVR and ANN had similar results. The LSTM is also more stable than the ANN as the deviation for the results over ten runs, is much smaller for the LSTM.

The improvement on the best performing benchmarked results is presented in Table 4-20. The LSTM in this study was compared to the STL Decomposition and ARIMA-ANN from the study by Gumani et al. [20]. There was an improvement 92.29% and 93.80%.

Table 4-19: Rossmann's Sales Dataset Results

	Train	**Test**			
	RMSE	**MSE**	**RMSE**	**MAE**	**MASE**
SVR	225.83	50499.18	224.72	177.47	0.09
ANN	361.33 ± 172	132985.09 ± 134 869	354.61 ± 163	280.67 ± 149	0.14 ± 0.07
LSTM	**30.93 ± 6**	**1086.38 ± 286**	**32.88 ± 4**	**22.55 ± 3**	**0.01 ± 0.001**

Table 4-20: Percentage Improvement on Best Benchmarked Results for Rossmann's Sales Dataset

	Literature RMSE	Actual RMSE	Δ RMSE	% Improvement
STL Decomposition [20]	426.4	32.878872	393.521128	92.29
ARIMA-ARNN [20]	530.4	32.878873	497.521127	93.80

4.4.2 Experiment 2

The results from Experiment 2 for the three optimal models can be seen in the tables below. It is noted that the results improve for the latter iterations as a larger portion of the dataset is used.

Table 4-21: SVR Experiment 2 Results for Rossmann's Sales Dataset

Split	MSE	RMSE	MAE	MASE
0	280513.259247	529.635025	425.577450	0.186067
1	83273.084577	288.570762	220.890384	0.098801
2	73917.510185	271.877749	223.373898	0.109857
3	263085.210603	512.918327	354.790828	0.180112
4	134986.603014	367.405230	303.239199	0.149720
5	104233.191708	322.851656	266.898530	0.131130
6	92938.974774	304.858942	240.809772	0.120102
7	229900.998848	479.479925	387.836456	0.196760
8	111028.179165	333.208912	267.742519	0.133076
9	63642.688528	252.275026	195.682200	0.097274

Table 4-22: ANN Experiment 2 Results for Rossmann's Sales Dataset

Split	MSE	RMSE	MAE	MASE
0	710476.836402	842.897880	693.163240	0.303059
1	681379.469241	825.457127	696.717528	0.311631
2	219841.254318	468.872322	370.768583	0.182346
3	584553.185083	764.560779	541.891727	0.275095
4	622916.657815	789.250694	620.693014	0.306459
5	115104.917949	339.271157	262.658524	0.129047
6	181972.919033	426.582840	341.961343	0.170551
7	264110.622013	513.916941	404.346286	0.205136
8	75839.443071	275.389620	227.301777	0.112975
9	38110.958792	195.220283	169.866762	0.084441

Table 4-23: LSTM Experiment 2 Results for Rossmann's Sales Dataset

Split	MSE	RMSE	MAE	MASE
0	120817.118325	347.587569	244.499339	0.106898
1	25104.931477	158.445358	113.579303	0.050802
2	8361.151020	91.439330	65.070482	0.032002
3	10308.788863	101.532206	66.548325	0.033784
4	3751.672594	61.250899	42.433358	0.020951

Split	MSE	RMSE	MAE	MASE
5	2898.424024	53.837014	42.998512	0.021126
6	1002.089548	31.655798	23.737336	0.011839
7	5223.972318	72.277052	48.778541	0.024747
8	876.960908	29.613526	21.225899	0.010550
9	1264.692752	35.562519	26.580532	0.013213

The following table summarises the results from Experiment 2 for the different algorithms. All three algorithms have mean values for the last 5 iterations, that are within a similar range to the results produced in Experiment 1. However, the LSTM has the lowest standard deviation and proves to be the most robust for this dataset.

Table 4-24: Summary of Experiment 2 Results for Rossmann's Sales Dataset (Last 5 iterations)

	MSE		RMSE		MAE		MASE	
	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev
SVR	120348.81	63865.39	338.53	84.73	271.79	71.17	0.14	0.037
ANN	135027.77	89665.04	350.08	124.87	281.23	92.87	0.14	0.048
LSTM	**2253.23**	**1849.21**	**44.59**	**18.20**	**32.66**	**12.39**	**0.02**	**0.006**

4.5 Global Energy Prediction Competition – Wind Dataset

4.5.1 Experiment 1

The Wind Power dataset is a large, non-linear dataset that displays a daily seasonality. The unit root test confirms that the dataset is non-stationary. Figure 4-5 shows the actual and predicted values for the Wind dataset.

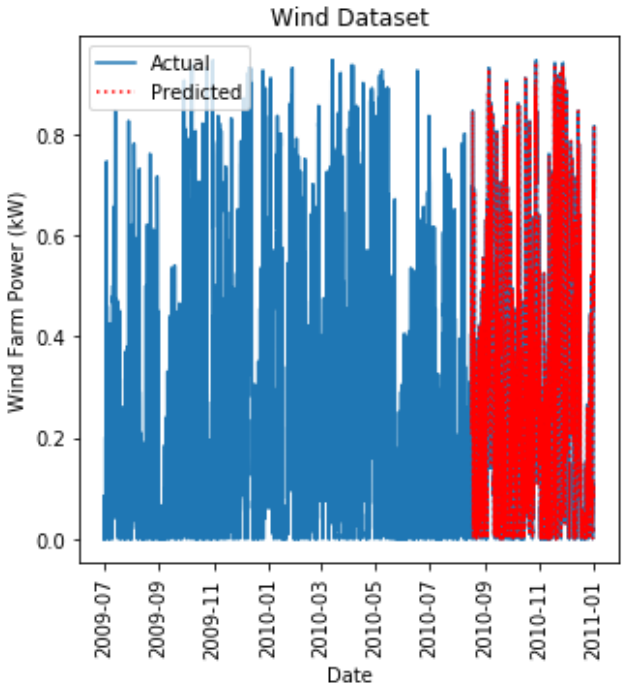

Figure 4-5: Wind Dataset

The optimal hyperparameter values found for the SVR model was C = 10, ε = 0.003, γ = 0.004. The optimal architecture found for the LSTM was a single layer with 80 nodes while for the ANN, four layers with 20, 50, 100 and 20 nodes were found to be optimal.

The performance results for the Wind dataset are presented in Table 4-25. The LSTM had the best training performance. However, the SVR outperformed the LSTM on the out of sample data by a small margin. The improvement on the benchmarked results is presented in Table 4-26. The SVR in this study showed an 98.96% improvement in RMSE over the KNN presented by Mangalova and Agafonov [37]. There was also an 98.95% improvement over Silva's [32] GBM.

Table 4-25: Wind Power Dataset Results

	Train	**Test**			
	RMSE	**MSE**	**RMSE**	**MAE**	**MASE**
SVR	**0.004**	**0.000002**	**0.002**	**0.001**	**0.024466**
ANN	0.004 ± 0.006	0.000019 ± 0.00008	0.003 ± 0.006	0.002 ± 0.003	0.058365 ± 0.06
LSTM	0.002 ± 4e-04	0.000003 ± 8 e-07	0.001 ± 2 e-04	0.001 ± 2 e-04	0.024499 ± 0.004

Table 4-26: Percentage Improvement on Best Benchmarked Results for Wind Dataset

	Literature RMSE	**Actual RMSE**	**Δ RMSE**	**% Improvement**
KNN [37]	0.1472	0.001529	0.145671	98.96
GBM [32]	0.14567	0.001529	0.144141	98.95

4.5.2 Experiment 2

The results obtained from Experiment 2 for the three optimal models can be seen in the tables below. At the initial iterations, the results are not in the same range as the results produced in Experiment 1 as only a portion of the data is used at that point. It is observed that all the models are robust as similar results are obtained when the size of the dataset is varied. The SVR is slightly more robust than the ANN and LSTM, as the initial iterations have similar results to the later iterations in comparison to the other two models. This is also reiterated by the standard deviation for the SVR, being much smaller than that of the ANN and LSTM as seen in Table 4-30. The table also shows that the mean of the last 5 iterations are similar to the results obtained in Experiment 1 for all three algorithms.

Table 4-27: SVR Experiment 2 Results for Wind Dataset

Split	MSE	RMSE	MAE	MASE
0	0.000069	0.008306	0.004658	0.086367
1	0.000013	0.003618	0.002556	0.056362
2	0.000006	0.002459	0.001906	0.041050
3	0.000003	0.001834	0.001511	0.032097
4	0.000005	0.002131	0.001721	0.036813
5	0.000004	0.001886	0.001457	0.029885
6	0.000002	0.001554	0.001215	0.024388
7	0.000002	0.001574	0.001249	0.025024
8	0.000002	0.001507	0.001218	0.024072
9	0.000002	0.001420	0.001133	0.022321

Table 4-28: ANN Experiment 2 Results for Wind Dataset

Split	MSE	RMSE	MAE	MASE
0	0.000015	0.003845	0.002984	0.065311
1	0.000007	0.002641	0.001943	0.045724
2	0.000009	0.002941	0.001856	0.042200
3	0.000013	0.003573	0.002841	0.063751
4	0.000013	0.003552	0.002776	0.062704
5	0.000006	0.002368	0.001940	0.041995
6	0.000006	0.002468	0.002066	0.043764
7	0.000017	0.004121	0.003417	0.072321
8	0.000004	0.002030	0.001621	0.033834
9	0.000008	0.002860	0.002226	0.046311

Table 4-29: LSTM Experiment 2 Results for Wind Dataset

Split	MSE	RMSE	MAE	MASE
0	0.000006	0.002391	0.001149	0.025504
1	0.000022	0.004684	0.003197	0.075783
2	0.000006	0.002361	0.001832	0.041862
3	0.000001	0.001220	0.000820	0.018453

Split	MSE	RMSE	MAE	MASE
4	0.000004	0.001995	0.001491	0.033765
5	0.000004	0.001902	0.001412	0.030648
6	0.000002	0.001403	0.000998	0.021191
7	0.000002	0.001581	0.001240	0.026292
8	0.000002	0.001448	0.000931	0.019469
9	0.000002	0.001347	0.001002	0.020884

The following table summarises the results from Experiment 2 for the different algorithms.

Table 4-30: Summary of Experiment 2 Results for Wind Dataset (Last 5 iterations)

	MSE		RMSE		MAE		MASE	
	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev
SVR	**0.000002**	**0.000001**	**0.0016**	**0.0002**	**0.0013**	**0.0001**	**0.03**	**0.003**
ANN	0.000008	0.000005	0.0028	0.0008	0.0023	0.0007	0.05	0.015
LSTM	0.000003	0.000002	0.0017	0.0004	0.0013	0.0004	0.03	0.009

4.6 Global Energy Prediction Competition – Load Dataset

4.6.1 Experiment 1

The Load dataset is a large, non-linear dataset that shows a daily seasonality. The KPSS unit root test shows that the dataset is non- stationary. The dataset is a multivariate dataset with two variables. The figure below shows the actual and predicted values for the dataset.

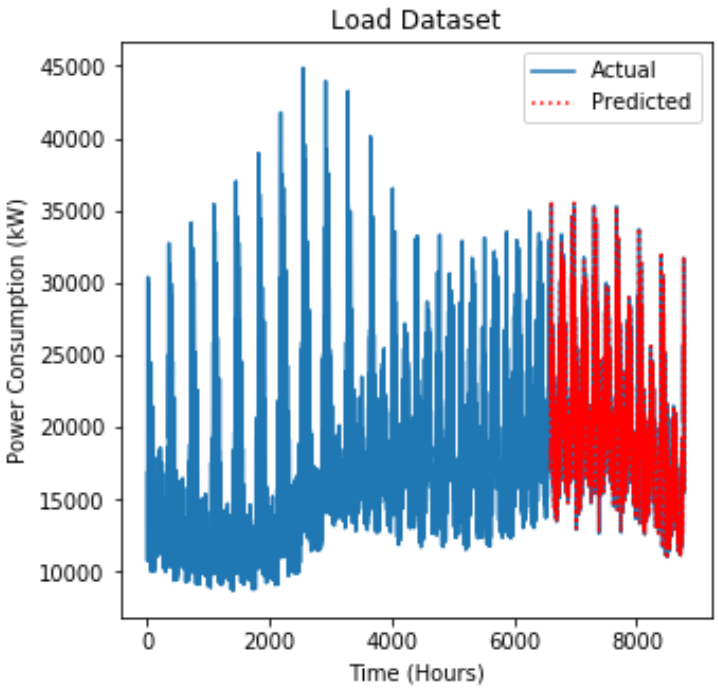

Figure 4-6: Load Dataset

The optimal hyperparameter values found for the SVR model was C = 13, ε = 0.008, γ = 0.01. The optimal architecture found for the LSTM was three layers with 80, 100 and 50 nodes respectively. For the ANN, five layers with 20, 50, 70, 50 and 20 nodes were found to be optimal.

The experiment results for the Load dataset are presented in Table 4-31. The LSTM outperformed the SVR by a slight margin. The LSTM has better stability than the ANN, as indicated by the smaller deviation from the mean in the results over 10 iterations.

The improvement on the benchmarked results is presented in Table 4-32. The RMSE of the LSTM in this experiment was 91.93% and 96.10% higher than the RMSE of the RNN and FFNN from the study by Busseti et al [52].

Table 4-31: Load Dataset Results

	Train	**Test**			
	RMSE	**MSE**	**RMSE**	**MAE**	**MASE**
SVR	176.46	10758.46	103.72	80.32	0.04
ANN	493.01 ± 557	123689.17 ± 827488	252.18 ± 723	185.90 ± 533	0.09 ± 0.26
LSTM	**71.88± 29**	**1926.67 ± 1652**	**42.78± 17**	**32.04± 12**	**0.02 ± 0.004**

Table 4-32: Percentage Improvement on Best Benchmarked Results for Load Dataset

	Literature RMSE	**Actual RMSE**	**Δ RMSE**	**% Improvement**
RNN [60]	530	42.778177	487.221823	91.93
FFNN [60]	1103	42.778177	1060.221482	96.10

4.6.2 Experiment 2

The results from Experiment 2 for the three optimal models can be seen in the tables below. The results from the initial iterations differ vastly from the results observed in Experiment 1, since only a small portion of the entire dataset is used for these iterations.

Table 4-33: SVR Experiment 2 Results for Load Dataset

Split	**MSE**	**RMSE**	**MAE**	**MASE**
0	269717.242375	519.343087	350.110474	0.190242
1	355625.644716	596.343563	368.356649	0.200144
2	189037.695928	434.784655	302.796647	0.156149
3	453143.290653	673.159187	402.917577	0.203155
4	287051.577668	535.771946	365.010846	0.186582
5	149097.076768	386.130906	283.718292	0.143989
6	126087.470096	355.087975	259.034858	0.130941
7	68807.351358	262.311554	199.705141	0.098847
8	44676.571745	211.368332	168.599389	0.082326

Split	MSE	RMSE	MAE	MASE
9	36378.420455	190.731278	160.831556	0.079368

Table 4-34: ANN Experiment 2 Results for Load Dataset

Split	MSE	RMSE	MAE	MASE
0	218752.369525	467.709706	312.519643	0.167243
1	584173.968361	764.312743	544.558979	0.293650
2	139778.294973	373.869355	258.648561	0.132712
3	25144.362458	158.569740	82.688298	0.041535
4	35320.757559	187.938175	146.097885	0.074456
5	38085.135718	195.154133	159.713130	0.080852
6	40098.588447	200.246319	126.383131	0.063749
7	28384.760498	168.477774	123.334713	0.060932
8	12715.547786	112.763238	98.897911	0.048210
9	81541.155120	285.554119	167.079795	0.082327

Table 4-35: LSTM Experiment 2 Results for Load Dataset

Split	MSE	RMSE	MAE	MASE
0	166421.725083	407.948189	322.435651	0.172550
1	64801.321574	254.561037	209.934197	0.113206
2	21914.054673	148.033965	114.064104	0.058526
3	1604.691426	40.058600	26.191675	0.013156
4	1339.035081	36.592828	28.836422	0.014696
5	1764.146238	42.001741	34.261731	0.017344
6	2063.712025	45.428097	34.422365	0.017363
7	2206.631601	46.974798	33.862638	0.016729
8	1209.864431	34.783106	27.473085	0.013392
9	1930.712518	43.939874	35.300995	0.017394

The following table summarises the results from Experiment 2 for the different algorithms. The RMSE mean seen in the last 5 iterations for the LSTM is similar to the results produced in Experiment 1. However, this is not the case for the SVR and ANN. The LSTM has a minimal standard deviation indicating that it is more robust than the SVR and ANN for this dataset.

Table 4-36: Summary of Experiment 2 Results for Wind Dataset (Last 5 iterations)

	MSE		RMSE		MAE		MASE	
	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev
SVR	85009.38	50121.99	281.13	86.44	214.38	54.73	0.11	0.029
ANN	40165.04	25537.05	192.44	62.57	135.08	28.08	0.07	0.014
LSTM	**1835.01**	**385.81**	**42.63**	**4.75**	**33.06**	**3.17**	**0.02**	**0.002**

4.7 Sterling Pound/United States Exchange Rate Dataset

4.7.1 Experiment 1

The Exchange rate dataset is a medium-sized, non-linear, univariate dataset. The unit root test confirmed that the dataset is non-stationary. The data has a cyclic trend and demonstrates seasonality as seen in the following figure.

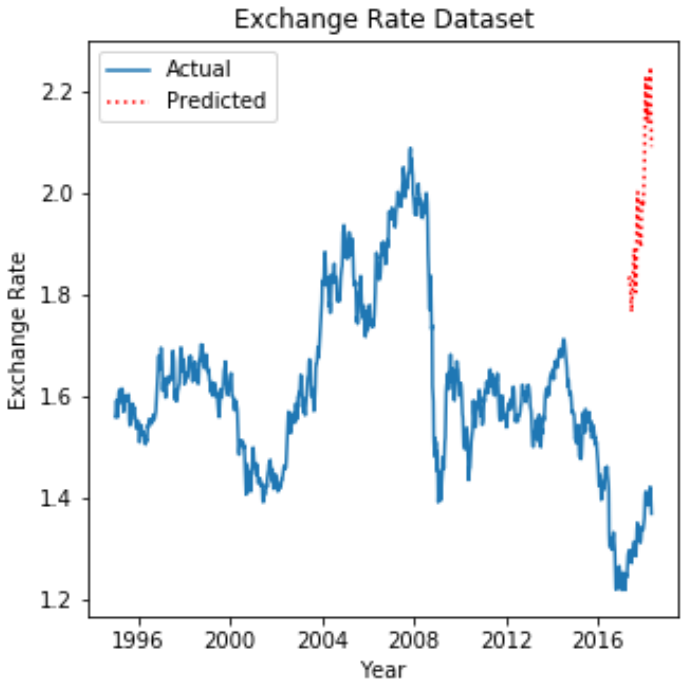

Figure 4-7: Exchange Rate Dataset

The optimal hyperparameter values found for the SVR model was C = 7, ε = 0.01, γ = 0.007. The optimal architecture found for the LSTM was three layers with 80, 100 and 20 nodes respectively. For the ANN, four layers with 20, 50, 100 and 20 nodes were found to be optimal.

The experiment results for the Exchange Rate dataset are presented in Table 4-37. The results are presented in the scale of the transformed data so that it is comparable to benchmarked studies. The ANN performed poorly for this dataset. The LSTM performed the best with an RMSE of 0.005. However, when considering the worst-case scenario of the LSTM, the SVR outperforms the LSTM. The LSTM is also seen to be more stable than the ANN.

The improvement on the benchmarked results is presented in Table 4-38. The SVR had an improvement of 31.47% over the ARIMA-ANN from Zhang's [21] study.

Table 4-37: Exchange Rate Dataset Results

	Train	Test			
	RMSE	**MSE**	**RMSE**	**MAE**	**MASE**
SVR	0.007	0.000037	0.006	0.005	0.61
ANN	0.023 ± 0.02	0.003150 ± 0.005	0.053 ± 0.034	0.05 ± 0.03 5	6.65 ± 5
LSTM	**0.003 ± 0.003**	**0.00002 ± 0.0003**	**0.005 ± 0.17**	**0.003 ± 0.006**	**0.49 ± 0.66**

Table 4-38: Percentage Improvement on Best Benchmarked Results for Exchange Rate Dataset

	Literature RMSE	Actual RMSE	Δ RMSE	% Improvement
ARIMA-ANN [21]	0,006602	0.004524	0.002078	31.47
ARIMA-ANN [30]	0.006135	0.004524	0.001611	26.26
ARIMA [16]	0.00561	0.004524	0.001086	0.19

4.7.2 Experiment 2

The Experiment 2 results for the three optimal models can be seen in the tables below. It is noted that the results are not always in the range of the results presented above. This is due to different data partition being used and demonstrates that the models are not robust when the partition sizes vary significantly from that used to tune the model parameters.

Table 4-39: SVR Experiment 2 Results for Exchange Rate

Split	MSE	RMSE	MAE	MASE
0	0.000078	0.008853	0.007340	0.689201
1	0.000058	0.007592	0.005952	0.775318
2	0.000030	0.005445	0.004426	0.603694
3	0.000311	0.017622	0.015408	2.183042
4	0.000079	0.008908	0.007383	1.028564
5	0.000159	0.012592	0.008840	1.254056
6	0.000029	0.005421	0.004510	0.581045
7	0.000018	0.004266	0.003541	0.455728
8	0.000021	0.004618	0.003598	0.475266
9	0.000154	0.012413	0.010601	1.423347

Table 4-40: ANN Experiment 2 Results for Exchange Rate

Split	MSE	RMSE	MAE	MASE
0	0.000272	0.016507	0.012456	1.169589
1	0.003706	0.060877	0.050832	6.621249
2	0.000411	0.020275	0.017408	2.374527
3	0.000211	0.014514	0.012386	1.754959
4	0.003163	0.056244	0.049673	6.920195
5	0.000413	0.020318	0.014733	2.089878
6	0.000044	0.006635	0.005352	0.689528

Split	MSE	RMSE	MAE	MASE
7	0.000038	0.006132	0.004920	0.633248
8	0.000338	0.018375	0.014091	1.861509
9	0.009739	0.098685	0.086403	11.600511

Table 4-41: LSTM Experiment 2 Results for Exchange Rate

Split	MSE	RMSE	MAE	MASE
0	0.000012	0.003469	0.002835	0.266207
1	0.000021	0.004553	0.003346	0.435906
2	0.000014	0.003735	0.003031	0.413496
3	0.000031	0.005536	0.004630	0.655982
4	0.000028	0.005302	0.004459	0.621266
5	0.000058	0.007634	0.006270	0.889355
6	0.000016	0.004013	0.003260	0.419970
7	0.000010	0.003220	0.002616	0.336695
8	0.000013	0.003642	0.002975	0.393048
9	0.000044	0.006634	0.005295	0.710906

The following table summarises the results from Experiment 2 for the different algorithms. The mean seen in the last 5 iterations for the algorithms are similar to the results produced in Experiment 1. However, the ANN has a high deviation indicating poor robustness. The LSTM is more robust than the SVR for this dataset.

Table 4-42: Summary of Experiment 2 Results for Exchange Rate Dataset (Last 5 iterations)

	MSE		RMSE		MAE		MASE	
	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev
SVR	0.0001	0.0001	0.0079	0.0043	0.0062	0.0033	0.84	0.464
ANN	0.0021	0.0043	0.03	0.0389	0.0251	0.0346	3.37	4.646
LSTM	**0.00003**	**0.00002**	**0.005**	**0.002**	**0.0041**	**0.0016**	**0.55**	**0.239**

4.8 Stock Market Indices Dataset

4.8.1 Experiment 1

The Stock Exchange dataset is a medium-sized, non-linear dataset that expresses a weekly seasonality. As seen in the following figure, the dataset demonstrates an upward trend. The unit root test shows that the dataset is non-stationary.

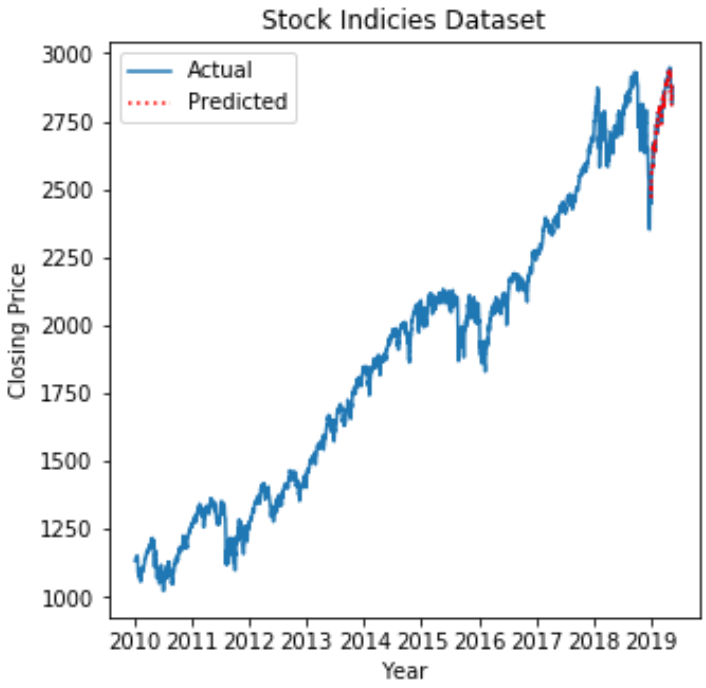

Figure 4-8: Stock Indices Dataset

The optimal hyperparameter values found for the SVR model was C = 7, ε = 0.002, γ = 0.008. The optimal architecture found for the LSTM was a single layer 500 nodes while for the ANN, three layers with 24, 50 and 100 nodes respectively, were found to be optimal.

The model performance for the Stock Exchange dataset is presented in Table 4-43. The LSTM outperformed the ANN and SVR. The LSTM model is also more stable than the ANN as indicated by the deviation of the results over ten runs.

The improvement on the benchmarked results is presented in Table 4-44. The LSTM presented in this study showed a significant improvement of approximately 96% for both the SVR and SVR hybrid in the study by Patel et al [63].

Table 4-43: Stock Indices Dataset Results

	Train	**Test**			
	RMSE	**MSE**	**RMSE**	**MAE**	**MASE**
SVR	6.42	47.00	6.86	5.03	0.64
ANN	10.13 ± 3	127.27 ± 70	11.16 ± 3	8.58 ± 3	1.09 ± 0.35
LSTM	**5.28 ± 0.27**	**42.32 ± 4**	**6.50 ± 0.3**	**4.70 ± 0.28**	**0.60 ± 0.036**

Table 4-44: Percentage Improvement on Best Benchmarked Results for Stock Indices Dataset

	Literature RMSE	**Actual RMSE**	**Δ RMSE**	**% Improvement**
SVR [63]	135.814	6.502987	129.311	95.21
SVR-SVR [63]	137.890	6.502987	131.387	95.28

4.8.2 Experiment 2

The results from Experiment 2 for the three optimal models can be seen in the tables below. It is noted that the results are not always in the range of the results presented above. This is due to different data partitions being used.

Table 4-45: SVR Experiment 2 Results for Stock Rate

Split	MSE	RMSE	MAE	MASE
0	508.485054	22.549613	19.344082	2.896846
1	54.207764	7.362592	6.039053	0.858349
2	348.415208	18.665884	12.691831	1.755872
3	175.366621	13.242606	10.619486	1.562375
4	98.100529	9.904571	8.186989	1.206560
5	59.567152	7.717976	5.681191	0.825774
6	47.252563	6.874050	5.619216	0.767038
7	189.520455	13.766643	12.125974	1.621739
8	127.148562	11.276017	9.213006	1.270228
9	121.776626	11.035245	8.863416	1.175260

Table 4-46: ANN Experiment 2 Results for Stock Rate

Split	MSE	RMSE	MAE	MASE
0	522.576687	22.859936	15.296632	2.290726
1	312.462841	17.676618	14.277923	2.029364
2	121.873693	11.039642	8.849069	1.224239
3	275.223079	16.589849	13.904059	2.045612
4	3144.595088	56.076689	52.961834	7.805267
5	398.065622	19.951582	13.256625	1.926881
6	232.931722	15.262101	11.258576	1.536826
7	234.842400	15.324568	12.859386	1.719827
8	195.903885	13.996567	9.849921	1.358042
9	476.758152	21.834792	15.761949	2.089984

Table 4-47: LSTM Experiment 2 Results for Stock Rate

Split	MSE	RMSE	MAE	MASE
0	13.529024	3.678182	2.917397	0.436891
1	31.083726	5.575278	4.711081	0.669600
2	62.416874	7.900435	6.534336	0.904003
3	44.926654	6.702735	5.860779	0.862258
4	41.149088	6.414755	5.377389	0.792495
5	38.330345	6.191151	4.448919	0.646661
6	32.035902	5.660027	4.233053	0.577823
7	114.206683	10.686753	9.699869	1.297270
8	102.395075	10.119045	7.796402	1.074916
9	85.631094	9.253707	6.906193	0.915739

The following table summarises the results from Experiment 2 for the different algorithms. The mean seen in the last 5 iterations for the algorithms are slightly higher than the results produced in Experiment 1. The standard deviation in the table below, indicates that the LSTM is slightly more robust than the SVR and ANN for this dataset.

Table 4-48: Summary of Experiment 2 Results for Stock Rate Dataset (Last 5 iterations)

	MSE		**RMSE**		**MAE**		**MASE**	
	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev
SVR	109.05	57.52	10.13	2.82	8.30	2.73	1.132	0.349
ANN	307.70	122.62	17.27	3.41	16.60	2.23	1.726	0.293
LSTM	**74.52**	**37.38**	**8.38**	**2.31**	**6.62**	**2.31**	**0.902**	**0.299**

4.9 Summary

A summary of the dataset characteristics is presented in Table 4-49. The characteristics that were identified was the size, linearity, stationarity, trend and seasonality.

Table 4-49: Dataset Characteristics

Dataset	**Size**	**Linearity**	**Stationarity**	**Trend**	**Seasonality**
Airline Passenger	Small	Non-linear	Non-stationary	Upward	Yearly
Sunspots	Small	Non-linear	Non-stationary	Cyclic	
Canadian Lynx	Small	Non-linear	Non-stationary	Cyclic	
Rossmann's Sales	Medium	Non-linear	Non-Stationary		Weekly
Wind	Large	Non-linear	Non-Stationary		Daily
Load	Large	Non-linear	Non-Stationary		Daily
Exchange Rate	Medium	Non-linear	Non-stationary		Weekly
Stock Market Indices	Medium	Non-linear	Non-stationary	Upward	Weekly

A summary of the MASE results for each algorithm and dataset is presented in Table 4-50. The table includes the MASE evaluation metric to show a comparison between the datasets, as the MASE metric is not dependent on scale while the MSE, RMSE and MAE are scale dependent. The objective is for the MASE to be as close to 0 as possible as this indicates a good algorithm. If the MASE is close to 1 or above 1 this indicates that algorithm performs poorly; and that the naïve forecast algorithm performs better [18].

The LSTM had the best performance across all datasets, except the Wind dataset. The ANN displayed the highest MASE across majority of the datasets. The LSTM and SVR MASE values differed by a small margin for the Canadian Lynx, Load, Wind and Stock Market Indices. When considering the LSTM's worst-case scenario for the Exchange Rate and Canadian Lynx datasets, the SVR outperforms the LSTM. It was also noted that although the LSTM's MASE value for the Canadian Lynx dataset, was slightly higher than that of the SVR, in terms of the remaining evaluation metrics the LSTM performed better. The SVR and ANN had similar results for the Sunspots and Rossmann's Sales dataset.

Table 4-50: MASE Results

Dataset	SVR	ANN	LSTM
Airline Passenger	0.364443	0.533355	**0.215912**
Sunspots	0.297071	0.296788	**0.070090**
Canadian Lynx	0.060749	0.333583	**0.061010**
Rossmann's Sales	0.088096	0.139317	**0.011195**
Wind	**0.024466**	0.058365	0.024499
Load	0.039578	0.091599	**0.015786**
Exchange Rate	0.611957	6.653813	**0.493488**
Stock Market Indices	0.637556	1.087479	**0.595379**

The table below summarises the stability of the different algorithms. The algorithms are ranked from most stable to least for each dataset. The SVR is the most stable since it is not a stochastic algorithm, while the LSTM is more stable than the ANN.

Table 4-51: Summary of the Algorithm Stability

	Most Stable	Second Most Stable	Least Stable
Airline Passenger	SVR	LSTM	ANN
Sunspot	SVR	LSTM	ANN
Canadian Lynx	SVR	LSTM	ANN
Rossmann's Sales	SVR	LSTM	ANN
Wind	SVR	LSTM	ANN
Load	SVR	LSTM	ANN
Exchange Rate	SVR	LSTM	ANN
Stock Market Indices	SVR	LSTM	ANN

A summary of the hyperparameters selected for the three algorithms is presented in Table 4-52. The hyperparameter values are presented for each dataset. The number of layers and number of neurons are presented in the table for the ANN and the LSTM. The hyperparameters for the SVR include the penalty parameter C, epsilon and gamma. The SVM algorithm was developed using the Python library, Scikit Learn, while the ANN and LSTM was developed using the Keras toolkit.

Table 4-52: Summary of Hyperparameters

	LSTM	**ANN**	**SVR**
Airline Passenger	(50) Batch Size = 20	(25 x 60 x 120 x 50 x 20) Batch Size = 20	C = 13, ε = 0.009, γ = 0.1
Sunspot	(50) Batch Size = 20	(10 x 30 x 20) Batch Size = 20	C = 2, ε = 0.002, γ = 0.1
Canadian Lynx	(50) Batch Size = 20	(10 x 50 x 100 x 20) Batch Size = 20	C = 13, ε = 0.008, γ = 0.1
Rossmann's Sales	(50) Batch Size = 20	(30 x 100 x 20) Batch Size = 365	C = 3, ε = 0.02, γ = 0.006
Wind	(80) Batch Size = 336	(20 x 50 x 100 x 20) Batch Size = 20	C = 10, ε = 0.003, γ = 0.004
Load	(80 x 100 x 50) Batch Size = 48	(20 x 50 x 70 x 50 x 20) Batch Size = 24	C = 13, ε = 0.008, γ = 0.01
Exchange Rate	(80 x 100 x 50) Batch Size = 52	(20 x 50 x 100 x 20) Batch Size = 20	C = 7, ε = 0.01, γ = 0.007
Stock Market Indices	(500) Batch Size = 365	(24 x 50 x 100) Batch Size = 365	C = 7, ε = 0.002, γ = 0.008

The following table summarises the results from Experiment 2. The algorithms are ranked from most to least robust for each dataset. The LSTM was the most robust for majority of the datasets while the SVR was robust for three of the datasets. In terms of the mean, there were four datasets where the SVR had a mean for the last five iterations that were closer to the mean in Experiment 1 than the ANN, while for the other four datasets, the ANN's mean in Experiment 2 was closer to that of Experiment 1.

Table 4-53: Summary of Algorithm Robustness

	Most Robust	**Second Most Robust**	**Least Robust**
Airline Passenger	SVR	ANN	LSTM
Sunspot	LSTM	SVR	ANN
Canadian Lynx	SVR	LSTM	ANN
Rossmann's Sales	LSTM	SVR	ANN
Wind	SVR	LSTM	ANN

	Most Robust	Second Most Robust	Least Robust
Load	LSTM	ANN	SVR
Exchange Rate	LSTM	SVR	ANN
Stock Market Indices	LSTM	ANN	SVR

Chapter 5

5 Discussion

This chapter discusses the results from the previous chapter as well as any insights gained. The evaluation metrics are discussed in the first section while a comparison between the experiment results and the results of the benchmarked studies are presented in the second section. The identified dataset characteristics that may have an impact on the algorithm performance are discussed in the third section followed by a comparison of the overall performance, stability and robustness of the machine learning algorithms.

5.1 Evaluation Metrics

The evaluation metrics that were predominantly used in the reviewed literature was the RMSE, MSE and MAE. The drawback of these evaluation metrics is that they are unit dependent. Therefore, these metrics are not suitable for evaluating and comparing model performance on different datasets, as the datasets consist of data with different scales. Literature that compared the performance of models across different datasets often normalised the data or used unitless evaluation metrics such as the MAPE.

For this study the data was transformed back to its normal scale for the analysis of the model performance. This was done in order to compare the results in this study to that of the benchmarked studies. This was done by comparing the RMSE. However, the RMSE could not be used to compare the results across different datasets. It was found that the MAPE and RMSPE were not suitable as the values were undefined or infinite for actual values that are close to zero. This is because the MAPE and RMSPE depend on the division of the actual values. Due to this dependency, the results from the MAPE and RMSPE can be unreliable as the results are skewed if the actual values are close to or are 0. The MASE overcomes this limitation as the dependency on the division by the actual values are removed. The MASE is also easy to interpret as compared to the RMSE and MSE, as the RMSE and MSE are based on the units of the data while the MASE values are typically between 0 and 1.

5.2 Experiment Results vs Benchmark Studies

The results from the experiment in this research were compared against the benchmark results found in Chapter 2. Chapter 4 presents the results of the comparison between the experiment results and the benchmarked results, in the form of a percentage improvement. The main objective was not to improve upon the results of existing research but to ensure that the results are at least within the same range so that an accurate comparison of the algorithms could be carried out.

The table below presents a summary of the results presented in Chapter 4. The table provides the percentage improvement of the best performing model in comparison to the best performing model in the reviewed literature.

Table 5-1:Summary of Percentage Improvement

Dataset	Percentage Improvement
Airline Passenger	46.41
Sunspot	88.30
Canadian Lynx	59.38
Rossmann's Sales	92.29
Wind	98.95
Load	91.93
Exchange Rate	0.19
Stock Market Indices	95.21

The results presented in this study showed an improvement over the reviewed literature. The difference in the results presented in this study and the results in the reviewed literature is due to multiple factors that are discussed below in detail.

The results for the Rossmann's Sales, Wind, and Load datasets showed a significant improvement over the benchmarked studies. This improvement is a result of the datasets being down sampled for the experiment as described in Section 3.3. The difference observed in the results for the Exchange Rate and Stock Indices datasets, is due the datasets being over a different time period than the datasets in the reviewed studies.

The overall differences seen between the results in this study and the reviewed literature is due to the feature engineering techniques used in this study, specifically the extraction of statistical features, as well as a more intensive hyperparameter tuning approach, which also plays a role in the significant improvement of accuracy. Performing additional data preprocessing, specifically normalisation of the data, also lead to improved performance in comparison to the reviewed literature.

The data partitioning approach used in this study for the SVR model validation, differed to the approach used in reviewed studies. While some studies used variations of cross validation, majority of the studies opted to use a hold-out method. This study opted for an iterative approach that allows for the optimal algorithm parameters to be selected, which increases the algorithm accuracy. Samsudin et al. [27] and Ismail and Shabri [38] used the k-fold cross validation method, without discussing the limitations of this method on time series data. The use of the k-fold cross validation

method compromises the results obtained during the algorithm validation step, resulting in sub-optimal parameters being selected for the algorithm.

5.3 Impact of Dataset Characteristics on Algorithm Performance

There were five key dataset characteristics that were identified in the exploratory data analysis. These characteristics are size, linearity, stationarity, trend and seasonality. Some reviewed literature highlighted these dataset characteristics for some of the benchmark datasets. The tests carried out in the exploratory data analysis of this research, confirmed the dataset characteristic insights found in the reviewed literature. However, the benchmark studies did not draw a clear comparison between the dataset characteristics and the model performance for all the benchmark datasets.

The characteristics were investigated further to determine if the algorithm performance was affected by the dataset characteristics. The datasets were categorised according to the key characteristics. The size of the datasets was classified as large, medium, and small. There were three datasets classified as small, two as large and three as medium. All the datasets were non-linear and non-stationary therefore, these characteristics were not taken into consideration. The trends were classified as cyclic, upward and none. Two datasets were classified as cyclic while another two datasets consisted of an upward trend. Three datasets presented weekly seasonality, two datasets demonstrated daily seasonality and one demonstrated a yearly seasonality. A discussion for each dataset characteristic is presented in the following sections.

5.3.1 Size

The relationship between the dataset size and algorithm performance is shown in Figure 5-1. All three algorithms had improved performance for larger datasets as these datasets allow for training over more data. This allows for a longer history for algorithms like the LSTM, to learn from the repetitive patterns of time series data. The LSTM benefits from the longer history due to its ability to retain information. The results seen in this experiment, verifies Min's [64] statement that the LSTM is better suited to data of longer periods. The ANN lacks the ability to retain information and therefore does not perform as well as the LSTM on the larger datasets. The findings of this experiment contradicts Min's [64] finding, that the ANN outperforms the LSTM on smaller datasets.

It is expected that the algorithms perform better for medium sized datasets in comparison to small datasets, but that is not observed in the results. This is because the medium sized datasets include the Exchange Rate and Stock Market Indices datasets. These datasets have poor performance across all three machine learning algorithms due to the nature of the data. These datasets' values are highly dependent on many external factors such as politics, global economic situations, resource prices and

international relationships, consequently it is challenging to forecast future values using univariate data.

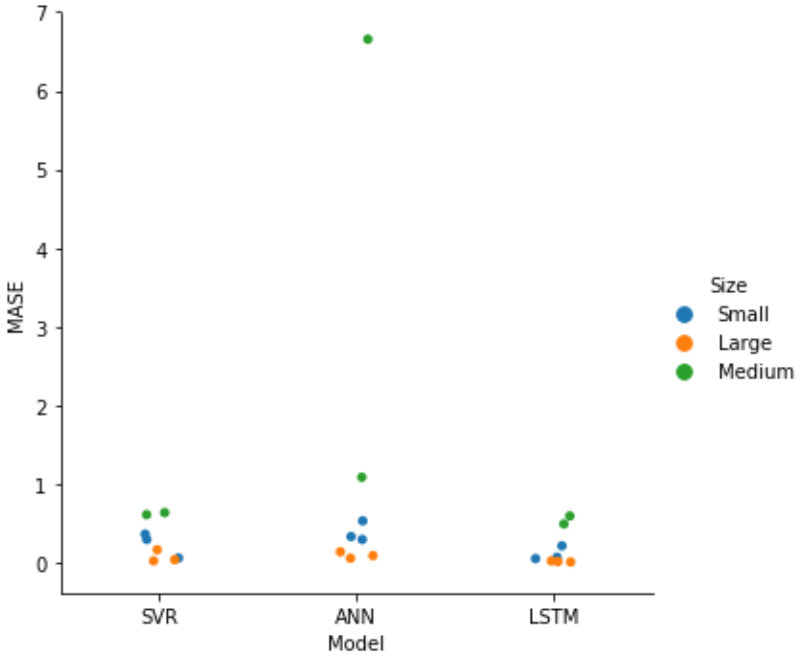

Figure 5-1: Relationship between Dataset Size and Algorithm Performance

5.3.2 Trend

The relationship between the dataset trend and algorithm performance is shown in Figure 5-2. The SVR performs better on datasets where there is no trend present. This confirms the findings of Crone et al. [65] that the SVR with a RBF kernel performs better on data without trends. The SVR in this experiment performed poorly on datasets with trend due to the shortfalls of the selected kernel as highlighted by Crone et al. [65]. The LSTM had the best performance for datasets with trend due to the algorithm's natural ability to retain information for long periods of time which allows the algorithm to model sequences and recurring patterns accurately. This confirms the findings in Min's [64] studies. However, the experiment shows that the LSTM provides similar performance for datasets without trends. This is because of the size of the datasets without trend are larger than that with trend.

The excellent results for the Rossman's Sales dataset are due to the dataset being multivariate, allowing the algorithm to take multiple factors into account when making a prediction. There is an outlier for the datasets with no trend, which is the Exchange Rate dataset. This is due to the external factors that affect this dataset not being taken into consideration for this experiment. As expected, the ANN performs well for datasets with no trend and is not deficient in scenarios where the

identification of sequences and patterns are not required for providing accurate forecasts on future values. The LSTM is superior in those cases where learning patterns and sequences are important.

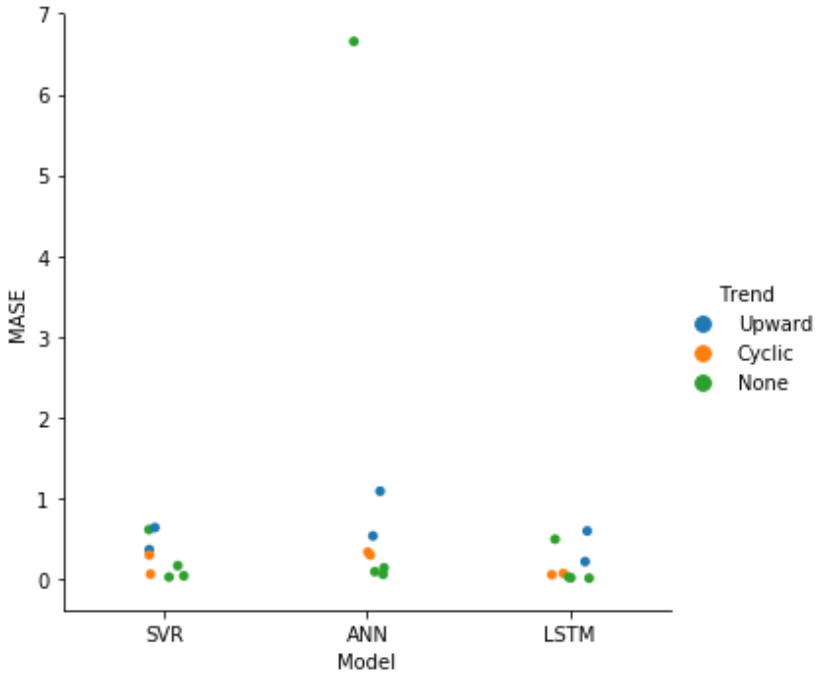

Figure 5-2: Relationship between Dataset Trend and Algorithm Performance

5.3.3 Seasonality

The relationship between the dataset seasonality and algorithm performance is shown in Figure 5-3. Traditional statistical time series prediction methods are known to benefit from the presence of seasonality in the data for providing more accurate predictions. Machine learning algorithms are expected to benefit in the same manner.

The results observed validate this hypothesis given that all algorithms used, SVR, ANN and LSTM, performed better in the presence of seasonality in the datasets, exclusive of the Exchange Rate and Stock Indices datasets which are considered exceptions due to their nature.

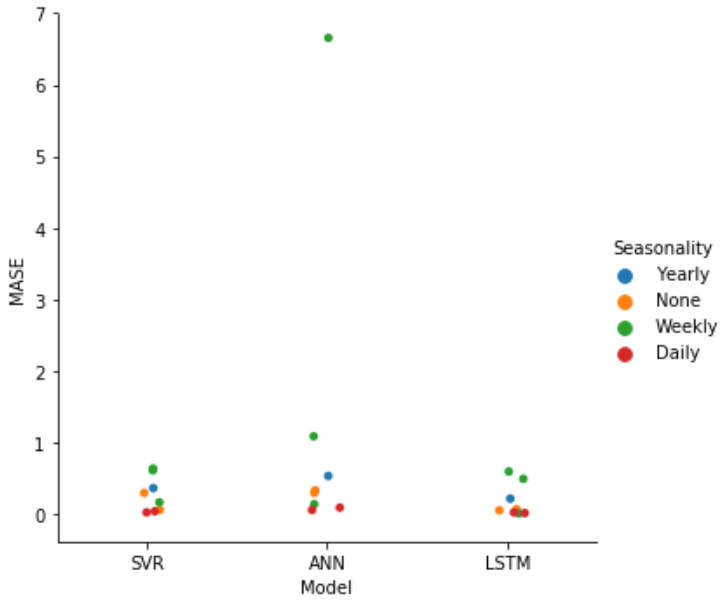

Figure 5-3: Relationship between Dataset Seasonality and Algorithm Performance

5.4 Comparison of Machine Learning Algorithms

5.4.1 Performance

A comparison of the MASE results for the various machine learning algorithms is shown in Figure 5-4. The ANN for the Exchange Rate dataset result was an outlier and was excluded from the figure for better visualisation of the performance. The SVR performed as effectively as the LSTM for some of the datasets.

The Rossmann's Sales dataset was multivariate and consisted of non-statistical features that captured the external factors that impacted sales such as public and school holidays, day of the week, whether the store was open or closed and the number of customers. A multivariate dataset coupled with an LSTM algorithm will capture these external factors and is expected to produce better accuracy.

The LSTM performs the best for time series due to its ability to handle sequence dependency with the use of its gated architecture that allows for the manipulation of memory. This ability allows the LSTM to perform well with datasets that consist of repetitive trends and seasonality.

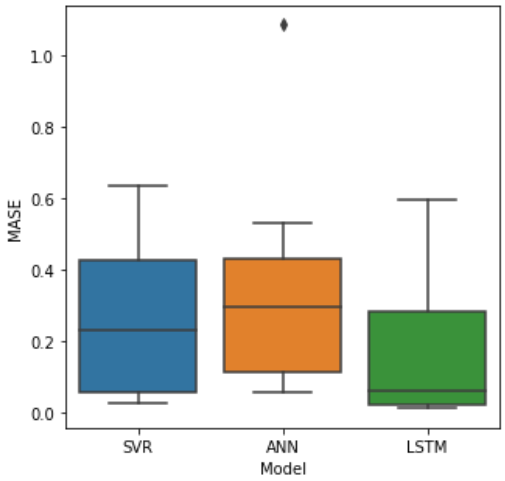

Figure 5-4: Comparison of Algorithm MASE Results

5.4.2 Stability

Due to the stochastic nature of the ANN and LSTM, the stability of these models is often a concern. Depending on the seed, the results for a model with the same parameters can differ since the model weights are randomly initialised every time the model is trained and the resultant weights across the neural network can marginally differ as a consequence of back propagation weight updates. The SVR does not have the issue of random initialisation and is therefore more stable than the LSTM and ANN since the same results are produced over multiple runs. Previous studies that used the eight datasets did not evaluate and compare of the stability of the algorithms tested.

For all the datasets the ANN was unstable. The worst-case scenario over ten runs differed vastly from the mean. As seen in Table 4-51, the LSTM proved to be more stable than the ANN. This was indicated by the deviation from mean over ten runs, which was minimal for the LSTM for majority of the datasets.

5.4.3 Robustness

It is expected that the SVR would be the most robust algorithm as the prequential method was used for hyperparameter tuning. However, as seen in Table 4-53, the LSTM proved to be the most robust algorithm for five out of the eight datasets while the SVR was the most robust for the remaining three datasets. The prequential method mimics a cross-validation approach and cross-validation increases the robustness of the model [34]. This is a consequence of hyperparameter tuning bias over a single validation dataset being eliminated by carrying out hyperparameter tuning and validating over different partitions of the data. If the prequential method is used for hyperparameter tuning, the robustness of the ANN and LSTM can be further improved. Previous studies that used the eight datasets did not compare the robustness of the algorithms tested.

The ANN was the least robust for majority of the datasets. Although, the data used in Experiment 2 was partitioned, it still maintained its temporal order. This was beneficial to the LSTM as it was able to use its manipulation of memory to detect the sequential patterns of the data allowing for relatively the same performance for all partitions of the data. Conversely, the ANN lacks this ability and was unable to accurately detect the patterns in the data partitions, resulting in outputs that deviated vastly from the mean. Although this relationship between time series data and the algorithm robustness has been highlighted, there was no correlation seen between the dataset characteristics discussed in Section 5.3 and the robustness of the model.

Chapter 6

Conclusion

The work in this study investigates and evaluates three machine learning algorithms for time series prediction. The study aimed to evaluate and compare the three machine learning algorithms for time series prediction problems for different types of datasets consisting of different characteristics. A critical analysis of the relationship between algorithm performance and dataset characteristics is also presented.

Some of the most predominantly used algorithms in the relevant literature for time series prediction problems include traditional statistical algorithms such as the ARIMA and its variants as well as machine learning algorithms including the SVR, ANN, LSTM and proposed hybrid algorithms. This study focused on machine learning algorithms at its simplest form, therefore the SVR, ANN and LSTM were investigated.

Experiments were carried out to investigate the performance of the aforementioned algorithms on eight datasets from different industries. It was found that LSTM performed the best overall due to its ability to handle sequence dependency. This algorithm characteristic allows for superior performance for datasets with trend and seasonality in comparison to the ANN and SVR. The SVR showed similar performance to the LSTM for some datasets. The performance of the ANN varied and showed poor stability on the time series data.

A time series dataset that has repetitive and sequence like behaviour is most desirable for the best performing algorithm, the LSTM. Additionally, a substantial amount of history is required as larger datasets lead to better accuracy for neural network algorithms.

Although the prequential method was used for the hyperparameter tuning of the SVR, the LSTM was the most robust for majority of the datasets. Due to the stochastic nature of the ANN and LSTM, the stability of the algorithms is often a concern. In addition to being the best performing algorithm, the LSTM was also more stable than the ANN. The stability and robustness of the ANN and LSTM can be improved by using the prequential method for splitting the data during parameter tuning.

Limitations of this study include dataset characteristics that adversely impact the results observed and the in-ability to perform experiments on much larger datasets because of limited processing power. Furthermore, the datasets each consisted of multiple dataset characteristics, biasing the results if some characteristics are more dominant than others. Future work could involve the isolation of dataset characteristics to allow for a more accurate representation of the relationship between a

specific dataset characteristic and algorithm performance. The benefits of multivariate datasets should also be further evaluated for time series predictions as it demonstrated improved results in the case of Rossmann's sales dataset when using an LSTM. The LSTM's ability to model inter-dependencies and co-relations between features in a multivariate dataset must be explored across all dataset characteristics.

References

[1] K. P. Ajoy and P. Dobrivoje, *Computational Intelligence in Time Series Forecasting - Theory and Engineering Applications*. London: Springer, 2014.

[2] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, no. 1, pp. 11–24, 2014.

[3] S. Mitchell, "The Application of Machine Learning Techniques to Time-Series Data," University of Waikato, 1995.

[4] R. Shumway and D. Stoffer, *Time Series Analysis and Its Applications: With R Examples*, 4th ed. 2009.

[5] R. K. Agrawal and A. Ratnadip, "An Introductory Study on Time Series Modeling and Forecasting," *Lambert Academic Publishing*, 2013.

[6] C. Chatfield, *Time-series forecasting*. New York, 2005.

[7] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255-, 2015.

[8] T. Fischer, C. Krauss, and A. Treichel, "Machine learning for time series forecasting - a simulation study," 2018.

[9] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. Massachusetts: The MIT Press, 2010.

[10] C. C. Aggrawal and D. S. Turaga, "Mining Data Streams: Systems and Algorithms," in *Machine Learning and Knowledge Discovery for Engineering Systems Health Management*, A. N. Srivastava and J. Han, Eds. Chapman and Hall/CRC Press, 2011.

[11] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Computer Science*, vol. 2, no. 160, 2021.

[12] K. H. Kouassi and D. Moodley, "An Analysis of Deep Neural Networks for Predicting Trends in Time Series Data," *SACAIR CCIS Springer proceedings*, 2021.

[13] H. Xu and S. Mannor, "Robustness and generalization," *Machine Learning*, vol. 86, no. 3, pp. 391–423, 2012.

[14] J. Brownlee, "How to Improve Deep Learning Model Robustness by Adding Noise," 2018.

https://machinelearningmastery.com/how-to-improve-deep-learning-model-robustness-by-adding-noise/ (accessed Apr. 29, 2021).

[15] J. C. B. Gamboa, “Deep Learning for Time-Series Analysis,” *Seminar on Collaborative Intelligence*, 2017.

[16] S. Masum, Y. Liu, and J. Chiverton, “Comparative analysis of the outcomes of differing time series forecasting strategies,” in *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD),* 2017, pp. 1964–1968.

[17] N. Davies and C. Chatfield, *The Analysis of Time Series: An Introduction*, 6th ed. New York, 2007.

[18] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed. Melbourne, Australia: Otexts, 2018.

[19] N. Sapankevych and R. Sankar, “Time series prediction using support vector machines: A survey,” *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24–38, 2009.

[20] M. Gurnani, Y. Korke, P. Shah, S. Udmale, V. Sambhe, and S. Bhirud, “Forecasting of sales by using fusion of machine learning techniques,” in *2017 International Conference on Data Management, Analytics and Innovation, (ICDMAI)*, 2017, pp. 93–101.

[21] P. G. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” *Neurocomputing*, vol. 50, pp. 159–179, 2003.

[22] C. H. Aladag, E. Egrioglu, and C. Kadilar, “Forecasting nonlinear time series with a hybrid methodology,” *Applied Mathematics Letters*, vol. 22, no. 9, pp. 1467–1470, 2009.

[23] A. Dingli and K. S. Fournier, “Financial Time Series Forecasting - A Machine Learning Approach,” *Machine Learning and Applications: An International Journal (MLAIJ)*, vol. 4, no. 1/2/3, pp. 11–27, 2017.

[24] S. B. Kotsiantis, “Supervised machine learning: A review of classification techniques,” *Informatica*, vol. 31, pp. 249–268, 2007.

[25] C. Crisci, B. Ghattas, and G. Perera, “A review of supervised machine learning algorithms and their applications to ecological data,” *Ecological Modelling*, vol. 240, pp. 113–122, 2012.

[26] S. Mullainathan and J. Spiess, “Machine Learning: An Applied Econometric Approach,” *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 87–106, 2017.

[27] R. Samsudin, A. Shabri, and P. Saad, “A Comparison of Time Series Forecasting using Support Vector Machine and Artificial Neural Network Model,” *Journal of Applied Sciences*, vol. 10, no. 11, pp. 950–958, 2010.

[28] M. Ghiassi, H. Saidane, and D. K. Zimbra, “A dynamic artificial neural network model for forecasting time series events,” *International Journal of Forecasting*, vol. 21, no. 2, pp. 341–362, 2005.

[29] K. Himakireeti and T. Vishnu, “Air Passengers Occupancy Prediction Using Arima Model,” *International Journal of Applied Engineering Research*, vol. 14, no. 3, pp. 646–650, 2019.

[30] M. Khashei and M. Bijari, “An artificial neural network (p, d, q) model for timeseries forecasting,” *Expert Systems with Applications: An International Journal*, vol. 37, no. 1, pp. 479–489, 2010.

[31] E. Yu, S. Lin, and X. Guo, “Forecasting Rossmann Store leading 6 month sales,” 2015.

[32] L. Silva, “A feature engineering approach to wind power forecasting: GEFCom 2012,” *International Journal of Forecasting*, vol. 30, no. 2, pp. 395–401, 2014.

[33] N. Charlton and C. Singleton, “A refined parametric model for short term load forecasting,” *International Journal of Forecasting*, vol. 30, no. 2, p. 364*368, 2014.

[34] V. Cerqueira, L. Torgo, and I. Mozetic, “Evaluating time series forecasting models: An empirical study on performance estimation methods,” *Machine Learning*, vol. 109, pp. 1997–2028, 2019.

[35] S. Yadav and S. Shukla, “Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification,” in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, 2016, pp. 78–83.

[36] M. Kraus, S. Feuerriegel, and A. Oztekin, “Deep learning in business analytics and operations research: Models, applications and managerial implications,” *European Journal of Operational Research*, vol. 281, no. 3, pp. 628–641, 2020, [Online]. Available: http://arxiv.org/abs/1806.10897.

[37] E. Mangalova and E. Agafonov, “Wind power forecasting using the k-nearest neighbors algorithm,” *International Journal of Forecasting*, vol. 30, no. 2, pp. 402–406, 2014.

[38] S. Ismail and A. Shabri, “Time series forecasting using least square support vector machine for canadian lynx data,” *Jurnal Teknologi*, vol. 7, no. 5, pp. 11–15, 2014.

[39] C. Wang and Y. Li, “Drug Store Sales Prediction,” Standford University, 2015.

[40] B. Ojemakinde, “Support Vector Regression for Non-Stationary Time Series,” University of Tennessee, 2006.

[41] R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[42] A. Borovykh, S. Bohte, and C. W. Oosterlee, “Conditional Time Series Forecasting with Convolutional Neural Networks,” 2017.

[43] W. He, “Load Forecasting via Deep Neural Networks,” *Procedia Computer Science*, vol. 122, pp. 308–314, 2017.

[44] G. Jin, F. Chu, and L. Wang, “Cancer Diagnosis and Protein Secondary Structure Prediction Using Support Vector Machines,” in *Support Vector Machines: Theory and Applications*, L. Wang, Ed. Berlin: Springer, 2005.

[45] N. K. Ahmed, A. F. Atiya, N. El Gayar, and H. El-Shishiny, “An empirical comparison of machine learning models for time series forecasting,” *Econometric Reviews*, vol. 29, no. 5, pp. 594–621, 2010.

[46] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “Statistical and Machine Learning forecasting methods: Concerns and ways forward,” *PLOS ONE*, vol. 13, no. 1, 2018.

[47] K. S. Shin, T. S. Lee, and H. J. Kim, “An application of support vector machines in bankruptcy prediction model,” *Expert Systems with Applications*, vol. 28, no. 1, pp. 127–135, 2005.

[48] C. F. Lin and S. De Wang, “Fuzzy support vector machines,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 464–471, 2002.

[49] V. Vapnik, *Statistical Learning Theory*. New York, 1998.

[50] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. Springer, 2000.

[51] X. Qiu, L. Zhang, Y. Ren, P. Suganthan, and G. Amaratunga, “Ensemble deep learning for regression and time series forecasting,” in *2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)*, 2014, pp. 1–6.

[52] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2014, pp. 338–342.

[53] Y. Bengio, P. Simard, and P. Frasconi, “Learning Long-Term Dependencies with Gradient Descent is Difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–165, 1994.

[54] A. Graves, A. R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” 2013.

[55] “Kaggle: Your Home for Data Science.” https://www.kaggle.com/ (accessed May 17, 2019).

[56] “Dataset Download.” https://vincentarelbundock.github.io/Rdatasets/datasets.html (accessed May 17, 2019).

[57] Y. Kajitani, K. W. Hipel, and A. I. Mcleod, “Forecasting nonlinear time series with feed-forward neural networks: A case study of Canadian lynx data,” *Journal of Forecasting*, vol. 24, no. 2, pp. 105–117, 2005.

[58] C. Jee and T. Singh, “Forecasting Rossmann Sales Figures,” Standord University.

[59] T. Hong, P. Pinson, and S. Fan, “Global Energy Forecasting Competition 2012,” *International Journal of Forecasting*, vol. 30, no. 2, pp. 357–363, Apr. 2014, Accessed: May 17, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207013000745?via%3Dihub.

[60] E. Busseti, I. Osband, and S. Wong, “Deep Learning for Time Series Modeling,” Standford University, 2012.

[61] “S&P 500 - Yahoo Finance.” https://finance.yahoo.com/quote/%5EGSPC/history?period1=1262296800&period2=1558044000&interval=1d&filter=history&frequency=1d (accessed May 17, 2019).

[62] F. E. H. Tay and L. J. Cao, “Improved financial time series forecasting by combining Support Vector Machines with self-organizing feature map,” *Intelligent Data Analysis*, vol. 5, no. 4, pp. 339–354, 2018.

[63] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Predicting stock market index using fusion of machine learning techniques,” *Expert Systems with Applications*, vol. 42, no. 4, pp. 2162–2172, 2015.

[64] J. Min, “Financial Market Trend Forecasting and Performance Analysis Using LSTM,” *ArXiv*, vol. abs/2004.0, 2020.

[65] S. F. Crone, J. Guajardo, and R. Weber, “A study on the ability of Support Vector Regression and Neural Networks to forecast basic time series patterns,” in *Bramer M. (eds) Artificial*

Intelligence in Theory and Practice. IFIP AI 2006. IFIP International Federation for Information Processing, vol. 217, Boston, MA: Springer, 2006, pp. 149–158.