Bachelor Thesis

Degree of Master of Science in Computer Science and Engineering, 300 credits

Dataset characteristics effect on time series forecasting:

comparison of statistical and deep learning models

Computer Science and Engineering, 15 credits

Halmstad 25/06 - 2023

Adam Ahlman & Adam Taylor

HÖGSKOLAN
I HALMSTAD

Abstract

Time series are points of data measured throughout time in equally spaced periods. They present characteristics such as level, noise, trend, seasonality, and outliers. Time series forecasting is the attempt to predict single or multiple future values. It holds significant relevance in numerous fields, including, but not limited to, healthcare, finance, and weather forecasting. It has recently gained more attention due to the COVID-19 pandemic, which highlighted the importance of predicting and managing crises. Two distinct methods of forecasting utilise either statistical or deep learning models, and the debate about the best model is still inconclusive. This thesis aimed to explicate the benefits and drawbacks of each approach pertaining to single-step and multi-step forecasting. The study applied four models, two of each method, on datasets of varying characteristics and measured their prediction accuracy and computing time. The prediction accuracy of each model was measured using commonly used evaluation metrics, including Root Mean Square Error. Subsequently, the results were compared with the features of the datasets to identify possible interconnecting relations between the factors. The findings concluded that the deep learning models generally produced a more accurate prediction but required more processing power and computing time. Contrastingly, the statistical models' predictions were less accurate but marginally faster. Furthermore, the forecast accuracy's most impactful characteristics were the dataset's trend and linearity. The code and datasets were published at: https://github.com/Adam20Taylor/BScThesis.

Sammanfattning

Tidsserier är punkter av data mätt under samma tidsintervall. De presenterar egenskaper så som nivå, brus, trend, säsongsvariation och avvikare. Tidsserieprognoser syftar till att försöka förutsäga ett eller flera nästkommande värden. Det har betydande relevans inom flera områden, inklusive, men inte begränsat till, sjukvård, ekonomi och väderprognoser. Det har nyligen fått mer uppmärksamhet på grund av COVID-19 pandemin vilket belyste vikten av att förutsäga och hantera kriser. Två metoder för förutspåendet är antingen genom statistiska- eller djupinlärningsmodeller och debatten om vilken modell som är bäst är ännu ofullständig. Huvudsyftet med denna uppsatts var att klargöra för- och nackdelar med de två tillvägagångsätten, med avseende på både enstegs- och flerstegprognoser. Studien gick ut på att undersöka fyra modeller, två från varje metod, och tillämpa dessa på datauppsättningar av varierande egenskaper. Modellernas beräkningstid mättes och deras prediktionsprecision utvärderades med hjälp av vanligen använda mått, som till exempel Root Mean Square Error. Resultaten jämfördes med datasetens egenskaper för att identifiera eventuella samband. Analysen visade att djupinlärningsmodellerna i allmänhet producerade noggrannare prognoser med nackdel av att de krävde mer processorkraft och beräkningstid. I kontrast var de statistiska metoderna marginellt snabbare men de gav mindre exakta svar. Vidare visade det sig att trend var den egenskapen som hade störst inverkan på prognosprecisionen. Koden och datauppsättningarna publicerades på: https://github.com/Adam20Taylor/BScThesis.

Table of contents

1. Introduction

Forecasting is making informed predictions or estimations of future events using historical and present data. It includes discovering and analysing dataset patterns, trends, and anomalies. It is used in many fields, such as weather forecasting, healthcare, finance, transportation, and supply chain management.

There are multiple models for producing forecasts, such as statistical, mathematical, and deep learning models. However, the discussion of what model to use in certain situations has yet to come to a clear conclusion. Therefore, it poses a problem to developers when deciding which method to employ when accounting for time and resources.

The effectiveness of statistical and deep learning models for time series forecasting will be compared in this research, along with the advantages and disadvantages of each approach. Additionally, the study strives to clarify when to use each model and how to select the most appropriate model given a forecasting problem.

1.1 Purpose

This project aims to compare and analyse the performance of several time series forecasting models. These models are divided into two groups: statistical and deep learning. This study will answer the research questions:

- How do different forecasting models compare when applied to varying datasets?
- What data characteristics affect the performance of different forecasting models?

The findings should aid in choosing the forecasting model depending on the dataset's characteristics.

1.2 Boundaries

The data used in this thesis will be univariate, meaning a single variable measured throughout time. The primary reasons for using univariate time series are the abundance and simplicity of data. The most common time series are univariate, while multivariate time series can be effectively reduced to univariate time series by excluding all but one variable. Other types of time series data contain more characteristics; these characteristics are generally complex and will, therefore, not be studied in this thesis.

2. Background

2.1 Description of time series

A time series is defined as points of data measured throughout time. The data points are commonly collected in equally spaced periods with a given sampling rate. Time series can have different characteristics, such as level, noise, trend, seasonality, and outliers. The expected value, or baseline of the time series, is called the level and is usually equal to the mean. Noise refers to random or unpredictable variations within the data. All measured data has some noise caused by measurement errors or sensor noise. Trend and seasonality are not always a part of a time series, but they significantly affect the different forecasting methods that can be used. Both trend and seasonality are recurring patterns within the data. The trend is the long-term upward or downward direction of the data. While seasonality is regular variations that occur over fixed time periods; for example, energy usage in Sweden rises during the winter months. Unusual events may also affect the data drastically and are therefore called outliers [1].

Time series that contain a trend or seasonality are called non-stationary time series. Conversely, a time series that does not include these characteristics are called stationary. Specific models only work on stationary datasets; others can handle stationary and non-stationary datasets. Non-stationary data can also be converted into stationary data using different methods. One method is first-order differencing which subtracts each value with the previous one and, therefore, can remove the trend within a time series [1, 2]. In Figure 1, a comparison of stationary vs non-stationary data is shown. The data on the left consists of random values between zero and ten; it is stationary because the values are randomly distributed around the level. In comparison, the data to the right has a clear trend and seasonality.

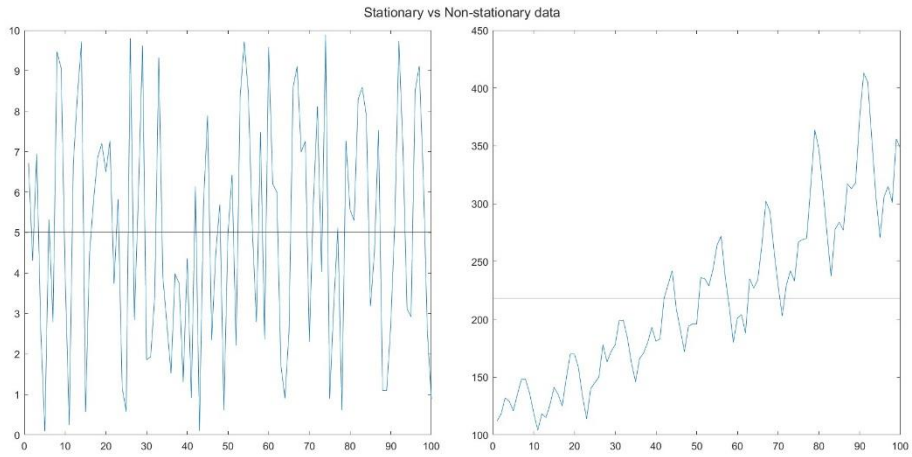

Figure 1: Comparison of a stationary (left) and non-stationary (right) dataset. The data points in the non-stationary dataset follow an incremental trend, while data points in the stationary dataset are randomly distributed around the level.

Linearity is not a characteristic of a time series, but it heavily impacts the performance of a forecasting model. All data points in a linear time series can be described as a linear combination of the previous values in the series. On the contrary, a non-linear time series does not have this property and can consequently be very complex. Real-world data is ordinarily a combination of both linear and non-linear parts [2].

2.2 Available models

2.2.1 Statistical models

Statistical modelling is fitting mathematical equations to datasets by adjusting the number of lags and the weights of the parameters. The lags refer to the number of past observations the model will consider when making a prediction. Therefore, statistical models can be defined as observing past data to predict future values. The models can be applied to datasets of relatively small sizes as it does not need any training data to make predictions. However, they are also fully operable on large datasets but limited to processing- time and power.

- The **autoregressive (AR) model** is one of the oldest prediction models. It is a simple mathematical formula that uses past data to predict future values. The general formulation of the AR model is described as follows:

$$AR(p):$$

$$Y_t = \Phi_0 + \Phi_1 X_{t-1} + \Phi_2 X_{t-2} + \cdots + \Phi_p X_{t-p} \quad (1)$$

  Where p defines the order of the model. The order establishes the number of Φ parameters and how many previous data points (lags) the

model uses to make a single prediction. The X_{t-p} represent the actual observed data point p time instances before the data point to be predicted. The Φ parameters can then be found using linear least squares [2, 3].

- The **Moving average (MA) model** was introduced a few years after the AR model [2]. As the name suggests, the MA model bases its predictions on the series' average. It is mathematically described as follows:

$$MA(q):$$

$$Y_t = \mu + \theta_1 E_{t-1} + \theta_2 E_{t-2} + \cdots + \theta_q E_{t-q} \quad (2)$$

$$E_t = X_t - Y_{t-1} \quad (3)$$

It works similarly to the AR model, where q represents the order of the model. However, instead of using a starting parameter and previous values, the MA model uses the mean represented by μ and the estimation errors E_t. The estimation errors are calculated by taking the difference between the observed value, X_t, and the previous prediction, Y_{t-1}. Furthermore, the fitting of the θ parameters must be found using more complicated methods such as maximum likelihood estimation or non-linear least squares [1].

- **Autoregressive Moving Average (ARMA) model** combines AR and MA. Hence, the mathematical formulation of ARMA is as follows:

$$ARMA(p, q):$$

$$Y_t = C + \Phi_1 X_{t-1} + \Phi_2 X_{t-2} + \cdots + \Phi_p X_{t-p} + \theta_1 E_{t-1} + \theta_2 E_{t-2} + \cdots + \theta_q E_{t-q} \quad (4)$$

Where p and q determine the order of the autoregressive and moving average terms, respectively. However, ARMA can only be applied to stationary datasets and is therefore not applicable in all situations. It can be expanded to ARIMA, which uses first-order differencing to account for trends and SARIMA, which also accounts for seasonality in the dataset [1, 2].

- **Exponential smoothening (ES)** methods use weighted averages of previous values in the series to perform their one-step predictions. The age of the data affects the weight exponentially, and therefore the newer data points are prioritised. The mathematical formulations of many different ES versions vary. However, equation 5 displays the fundamental formulation, simple exponential smoothing.

$$ES(W):$$

$$Y_t = W \times X_t + (1 - W) \times Y_{t-1} \quad (5)$$

The weight used in the model is represented as W, and X and Y represent the observed values and predictions, respectively [1, 4].

Two of the most well-known expansions of ES are Holt's linear trend method and Holt-Winters' seasonal method, commonly referred to as double ES and triple ES correspondingly. The double ES model has two smoothing equations and can therefore factor in the trend in its prediction. Similarly, the triple ES model uses three smoothening equations to calculate its output. However, triple ES has two variations, the additive and multiplicative methods. Both these variations take the level, trend and seasonality into account when forecasting and can, therefore, be applied to non-stationary data [1].

2.2.2 Deep learning models

Deep learning is a particular form of neural networks. Neural networks consist of layers of interconnected nodes that process information based on inputs and produce outputs. They can be trained on non-linear datasets of differing sizes to recognise patterns and hidden relationships in the data. Thus, making them applicable in time series analysis and forecasting [5].

The models train on the datasets and compares the predicted value to the actual one and corrects itself through optimization algorithms such as Adam or gradient descent. They iterate over the same dataset multiple times and a completion of the whole dataset is called an Epoch. The optimal number of epochs required to achieve the best outcome varies across datasets; however, an excessive number of epochs can lead to a phenomenon known as overfitting. As a result, the model becomes overly acclimated to a specific dataset and performs poorly when applied to real-world data.

Compared to statistical models, deep learning forecasting models were developed to forecast complex non-linear time series better [2]. However, the models require many training data points to produce accurate predictions. Deep learning models can also be harder to interpret as they only perform their assigned task while giving limited information about how the result was calculated.

- **Recurrent Neural Networks (RNN)** are ubiquitous in time series prediction. To learn the characteristics of the time series and then estimate future values, an RNN does a recurrent analysis of a set of historical data. The recurrent analysis utilises internal states between each time variable to create a model of the functional relationships within the data, to predict the future of a data sequence. Graphical visualisation of the described structure can be seen in Figure 2. The

input, output and internal state have weights between each other, indicated by U, W and V [2, 6].

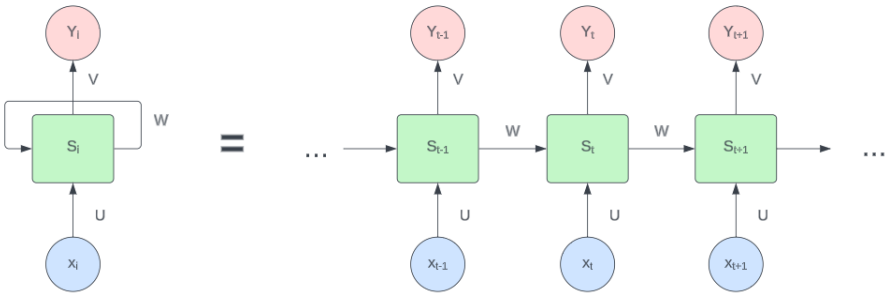

Figure 2: Structure of RNN. Performs recurrent data analysis and predicts future values with the help of internal states. X is the input, S is the inner state, and Y is the output.

A simple RNN employs the equations seen in equations 6 and 7. The weights, and the bias vectors, characterised by b_s and b_y, are learnable terms the neural network uses to fit the data better. The x_t refers to the input and y_t is the output, which in context of time series forecasting stands for the observed and predicted value respectively. When creating a neural network, the activation functions must be specified, represented by f and g [6]. The most frequently used activation functions for RNNs are called the sigmoid (σ), hyperbolic tangent (tanh) and the rectified linear unit (ReLU) functions.

$$S_t = f(U \times x_t + W \times S_{t-1} + b_s) \quad (6)$$

$$y_t = g(V \times S_t + b_y) \quad (7)$$

The main problem with simple RNNs is that it forgets past data; this problem is called the vanishing gradient problem [2, 7]. The problem makes the model unable to capture long-term dependencies and fails to account for patterns that extend across a significant portion of the dataset.

- **Long Short-Term Memory (LSTM)** is a more developed version of RNN and was created as a partial solution to the vanishing gradient problem. LSTM changes the calculation of the internal state and incorporates an additional state called a cell state. Consequently, it can remember long-term dependencies. LSTM also uses gates for the information's removal, multiplication, and addition. These gates are called the input, output, and forget gates. The state calculations within LSTMs can be mathematically formulated with equations 8-13 [2, 6].

$$i_t = \sigma(U_i \times x_t + W_i \times s_{t-1} + b_i) \quad (8)$$

$$o_t = \sigma(U_o \times x_t + W_o \times s_{t-1} + b_o) \quad (9)$$

$$f_t = \sigma\big(U_f \times x_t + W_f \times s_{t-1} + b_f\big) \quad (10)$$

$$\tilde{c}_t = \tanh(U_c \times x_t + W_c \times s_{t-1} + b_c) \quad (11)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (12)$$

$$s_t = \tanh(c_t) \odot o_t \quad (13)$$

The input, output and forget gates equations are denoted as i_t, o_t and f_t, respectively, while s_t and c_t correspond to the internal state and cell state. It is necessary to calculate the candidate state, denoted by $\tilde{c}_t$, before calculating the cell state. The equations of the gates and candidate state follow the same structure as the state calculation for the simple RNN, seen in equation 6. The $\odot$ in equations 12-13 corresponds to an element wise vector multiplication. Unlike simple RNNs, which require user-defined activation functions, LSTM typically uses two specific activation functions: the sigmoid (σ) and the hyperbolic tangent (tanh) functions.

- The **Gated Recurrent Unit (GRU)** model also attempts to solve the vanishing gradient problem. GRU is similar to LSTM but uses fewer gates, called update and reset gates, in slightly different ways. Consequently, GRU requires fewer parameters than LSTM but still achieves comparable results. GRUs can be described mathematically with the following equations [6, 8].

$$z_t = \sigma(U_z \times x_t + W_z \times s_{t-1} + b_z) \quad (14)$$

$$r_t = \sigma(U_r \times x_t + W_r \times s_{t-1} + b_r) \quad (15)$$

$$\tilde{s}_t = \tanh(U_s \times x_t + W_s(r_t \odot s_{t-1}) + b_s) \quad (16)$$

$$s_t = (1 - z_t) \odot s_{t-1} + z_t \odot \tilde{s}_t \quad (17)$$

The gate equations, z_t and r_t, also follow the same structure as the state equation for simple RNNs, seen in equation 6. Whilst the equations for the candidate state and internal state, denoted as $\tilde{s}_t$ and s_t, illustrate how the utilisation of the gates in GRUs varies from that of LSTM. For example, in equation 16, the reset gate, r_t, is multiplied element wise by the state of the previous time instance and in equation 17, the update gate, z_t, is used for both terms in the addition.

2.3 Analytical tests

Analytical tests can be applied to prove the existence of specific properties and characteristics within the data. Stationarity is an example of data characteristics that can be determined, which is essential for choosing the most optimal expansions of the statistical forecasting models. Furthermore, analytical tests can also be used to find a baseline in the choice of parameters for the ARMA-based models.

2.3.1 ACF and PACF

Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) are statistical tests used to find correlations between lags (previous values) and their respective time series value. Both functions help determine a statistical model's order by observing the number of points outside the threshold.

The ACF measures the linear correlation between a time series and its lagged values at different time lags. The correlation coefficients between the y-axis and the time lag between the x-axis are shown in the ACF graph. The ACF is used to determine if there are significant patterns or seasonal variations in the data [1]. The ACF reveals explicitly that the series has a strong linear relationship with its initial value, suggesting the presence of a trend in the data if it exhibits a significant correlation at the initial lag (lag 1). The presence of seasonality or cyclical patterns in the data indicates whether the ACF has a strong correlation over several lags.

The PACF accounts for the impact of intermediate delays and assesses the correlation between a time series and its lagged values. The PACF figure contrasts the time delays on the x-axis with the partial correlation coefficients on the y-axis [9]. The PACF helps determine the relationship between a time series model's moving average (MA) and autoregressive (AR) components. Notably, the PACF only exhibits statistically significant correlations up to the model's AR terms' order. Suppose the PACF substantially correlates at the initial lag (lag 1). In that case, it signifies that after accounting for the impact of any intermediate delays, the series has a robust linear connection with its initial value, suggesting the presence of an AR (1) model. This means the model will account for a single previous value in the prediction.

2.3.2 Augmented Dickey-Fuller test

One of the most common statistical tests used to determine whether a time series is stationary is called the Augmented Dickey-Fuller (ADF) test. ADF is a so-called unit-root test, which determines if a unit root exists within the data. A unit root refers to a characteristic of a time series variable where the variable's mean or average value tends to persist or drift over time. If a time series contains a unit root, the series is non-stationary. The null hypothesis for the test is that there exists a unit root in the data, and the alternative hypothesis is that there is no unit root. This means the time series is stationary if the p-value, the test result, is lower than the specified significance level, typically 0.05 [10].

2.4 Similar projects

Similar comparative studies have been made to compare these models [11-13]. In “A comparison of ARIMA and LSTM in forecasting time series” [13], financial data of twelve different stocks were used to compare ARIMA and LSTM. Their results indicated that LSTM outperformed ARIMA by 85% but did not discuss the data attributes that may have influenced the model’s performances. In contrast, a similar study called ”A comparison between arima, lstm, and gru for time series forecasting” [12] applied ARIMA, LSTM and GRU on a time series of bitcoin prizes and found that ARIMA outperformed both deep learning models. It is unclear why these studies got contradicting results while using datasets of similar characteristics and sizes.

3. Methodology

3.1 Choice of models

3.1.1 Deep learning methods

Recurrent neural networks (RNN) and Convolutional Neural Networks (CNN) are both neural networks, but their focus differs. Although CNN can be used for time series analysis using a sliding window approach to identify temporal dependencies, its primary focus relies on image analysis and recognition [14]. RNN is used primarily on sequential data, and its advantage of handling long-term dependencies on such data is why the RNN model was chosen over CNN.

Even though RNN is an effective model for predicting time series, it has two significant limitations. Firstly, they cannot process input sequences of varying lengths due to their fixed memory size and suffer from the vanishing gradient problem. The problem recedes, as stated before in the model, in forgetting past values. A solution to the difficulty is utilising gating mechanisms; as stated in 4.2.2, both LSTM and GRU do. The two models can also handle variable-length input sequences, which solves the first problem. The model's ability to solve both issues is why both are employed in this study.

3.1.2 Statistical methods

Exponential smoothening is considered state-of-the-art in many different branches of time series forecasting. It is commonly used to forecast retail demand and is helpful for inventory management. As stated in 2.2.1, it can account for trend and seasonality with the help of two expansions. Its flexibility and simplicity are the rationales behind utilising the model for this research.

An additional state-of-the-art model is ARIMA which can handle datasets of a wide range. It can provide accurate predictions even with noisy and missing data, making it a good choice for real-world data. Therefore, it is widely used within finance and economics, making it an appropriate model for this study.

3.2 Types of implementations

To make the comparison between the prediction results of the models fairer, it is necessary to place restrictions on the forecasting process. One such limitation is to allow only single-step forecasting to be performed by the models, meaning that the models can only provide predictions for the next single period. Commonly, single-step forecasting is implemented through a rolling window implementation. This approach helps to avoid any biases that may arise from variations in the length or scope of the forecasting horizon,

making the comparison of the model results more objective. However, multi-step forecasting is needed to determine the correlations between characteristics and forecasting accuracy. Therefore, both single- and multi-step forecasting were implemented.

3.2.1 Single-step forecasting

To compare the performance of each model effectively and accurately, a rolling window version of each model was implemented. The algorithm uses a fixed-size window of historical data to make a prediction, then move the window one step forward in time and repeats the process to obtain a sequence of predictions. The window size determines the number of lags that will be included to make the prediction. This makes the comparison fairer because all models get access to the same number of data points for each prediction.

The ARMA model's order determines the number of data points the model can use for each prediction. Therefore, the order of the ARMA model was the determining factor of the window size for the other models. Through ACF and PACF plots and trial and error, the p and q values for the ARMA model were set to three for all datasets. Consequently, the window size for the other models was also selected to three. Furthermore, the window size had an insignificant effect on the prediction accuracy when it was increased to larger values.

3.2.2 Multi-step forecasting

As previously mentioned, multi-step forecasting is vital to find correlations between the characteristics of the data and the forecasting performance. It accounts for the whole dataset when making multiple future predictions. The data that the model use persists the same throughout the forecasting process. Since this implementation is widely used in the industry, many implementations and libraries are available to support its use.

3.3 Python libraries for model implementations

All models were implemented using established Python libraries. The Statsmodels library was used for the implementation of the statistical models. Many statistical models and tests are integrated into this library, and it operates well with the Pandas library, which was used to read the CSV files containing the data [15].

For the deep learning models, both the Keras and PyTorch libraries were used. Keras was used to implement the single-step versions of LSTM and GRU, while PyTorch was used to implement the multi-step versions.

3.4 Datasets

Several datasets with varying characteristics and sizes were used, and most of them were acquired using public resources on the Internet. The table below presents the datasets and their characteristics and properties.

Table 1: Table visualising the different characteristics, linearity and sizes of each dataset used in this study

Datasets	Airline passengers	Total COVID-19 cases	Machine temperature	Gothenburg temperature
Trend	Additive	Additive	No trend	No trend
Seasonality	Significant	Slight	Non-seasonal	Significant
Linearity	Partially linear	Mostly linear	Non-linear	Non-linear
Number of data points	144	1 182	41 140	180 591

- The airline passenger dataset is commonly used in time series forecasting. As shown in Figure 3, the dataset has a simple, additive trend and a clear seasonality period of 12, which makes the dataset ideal for time series forecasting.

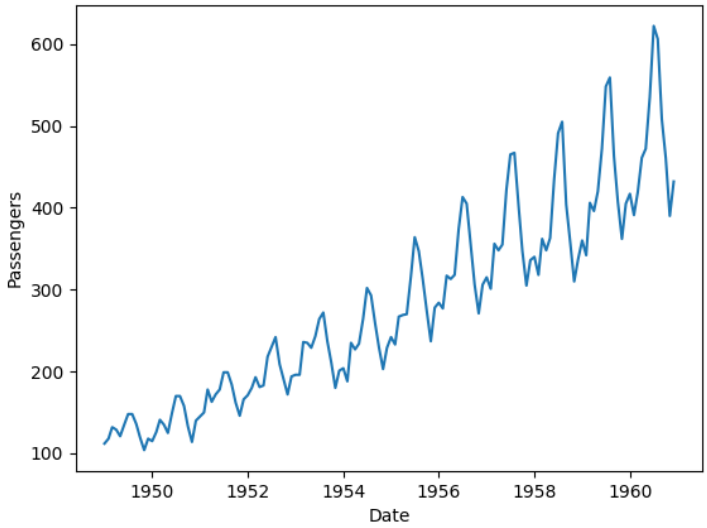

Figure 3: Plot visualising the airline passengers dataset, which clearly shows a trend and seasonality with a seasonality period of 12 months

- The dataset containing the total number of COVID-19 cases worldwide is available to the public and may be obtained on "Our World in data" [16]. The dataset was chosen for its strong trend and linearity. The ACF plot found a slight seasonality within the dataset with a period of seven. The dataset is plotted in Figure 4, which clearly illustrates the trend, but due to its little influence on the data, the seasonality is not visible.

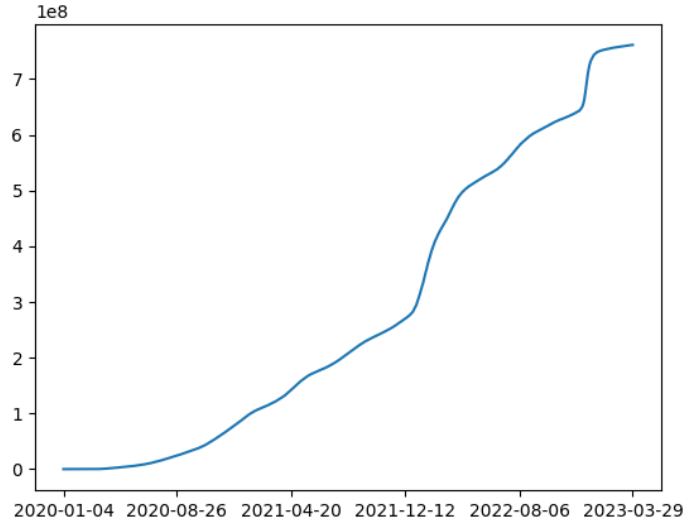

Figure 4: Total global COVID-19 cases show a strong trend, rising from a single point in January 2020 to 760 million in March 2023.

- An industrial partner provided the machine temperature dataset. As a result, this dataset cannot be distributed further and is, therefore, not a part of the published material. The dataset comprises an AR (40) model output applied to a machine's observed internal temperature. Statistical tests were run on the dataset, and it was evident that it was stationary and therefore lacked any overarching trend or seasonality. Figure 5 shows that the dataset exhibits random local upward and downward trends throughout the entire dataset.

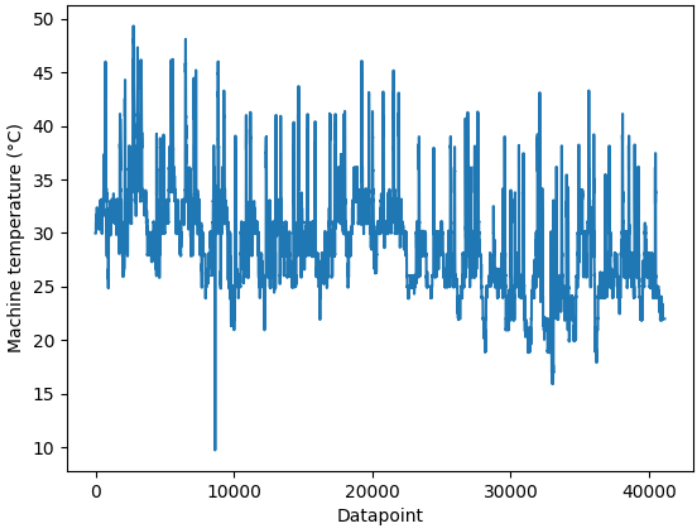

Figure 5: Plot of the internal temperature of a machine provided by an industrial partner. It shows no consistent seasonality or trend. However, random local trends are shown.

- The Swedish Meteorological and Hydrological Institute (SMHI) provides open weather data from weather stations around Sweden

[17]. The seasonality of the outside temperature is inherent and two-fold; It has a monthly and a daily seasonality period. An upward trend was anticipated as well. However, the statistical tests found no evidence of a long-term trend over the entire dataset. The selected weather station in Gothenburg began recording weather data in 1961. The current measurements are taken once every hour, whereas the older ones are collected thrice daily. Thus, the dataset was reduced to only include the data points from 2001 to 2022. Figure 6 visualises the condensed version of the SMHI dataset.

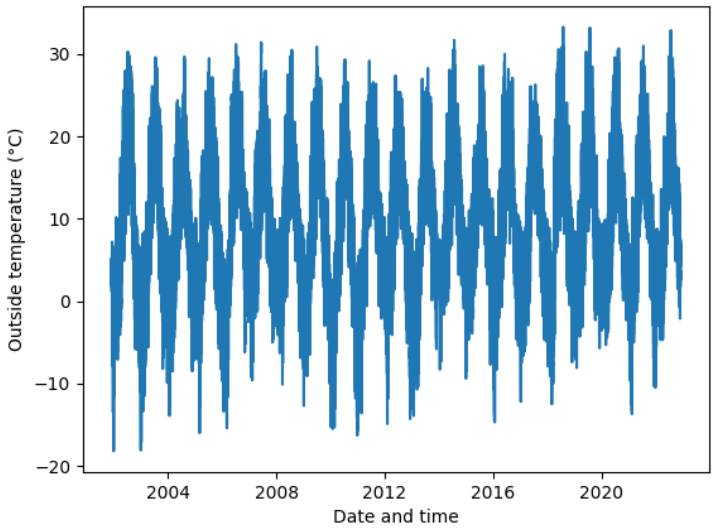

Figure 6: Plot of the cut-down version of the SMHI dataset. It shows the outside temperature in Gothenburg between the years 2001 and 2022. Both the daily and monthly seasonality is shown clearly.

3.5 Evaluation metrics

The forecasts from the chosen models were evaluated depending on their RMSE, MAPE and MAE values. Forecast accuracy is commonly measured and compared using these methods. Previous studies have claimed that the grading produced by these methods is tough to interpret individually [18]. However, due to the comparative nature of this study, individual accuracy grading is less critical. Instead, the difference between the forecasting models and datasets is in focus. Therefore, using more commonly used accuracy gradings is more important than the clarity of the grading.

RMSE is a regression analysis that aids in understanding the relationship between the output variables and one or more predictor variables. It stands for Root Mean Square Error and is a metric that shows the average distance between the predicted and actual values. A lower value means a superior fit for the model. The mathematical equation is shown in equation 18.

$$\sqrt{\sum_{i=1}^{n} \frac{(y_i - x_i)^2}{n}} \quad (18)$$

The y_i represents the observed, actual value of the i^{th} observation. While the x_i is, the i^{th} predicted value derived from the prediction model. The n is the sample size of observations. Significant inaccuracies are amplified by the squaring component in the equation, which means singular outliers in prediction accuracy significantly affect the grading [19].

MAE is an abbreviation for Mean Absolute Error. It is used on regression models and measures prediction accuracy for a forecasting model. MAPE stands for Mean Absolute Percentage Error, and as the name suggests, it produces a percentage based on the value of MAE. The following equations can calculate the MAE and MAPE scores, where MAE is equation 19, and MAPE is equation 20. [18, 19].

$$\frac{1}{n}\sum_{i=1}^{n} |y_i - x_i| \quad (19)$$

$$\frac{1}{n}\sum_{i=1}^{n} \frac{|y_i - x_i|}{y_i} \quad (20)$$

Like the RMSE, the y_i stands for the observed, actual value of the i^{th} observation. While the x_i is, the i^{th} predicted value derived from the prediction model. The n is the sample size of observations. One flaw of the MAPE score is when the observed value is equal to or close to zero. This flaw causes scores larger than 100% and is, therefore, impossible to interpret.

These gradings were used to measure the difference in prediction accuracy for each dataset. The differences were then further examined by utilising the datasets to discover overarching relationships between the attributes of the data and the prediction accuracy of each model. The main features within the data that were analysed were size, linearity and stationarity.

4. Results

4.1 Single-step forecasting

Table 2 showcases the result of the single-step forecasting. In general, the statistical models attained a greater numerical value in terms of the evaluation metrics, indicating a diminished accuracy in their predictions. However, the deep learning models were severely slower than the statistical models.

Table 2: Table showcasing the results from the single-step forecasting methods. The previously mentioned evaluation metrics, the computation time and the epochs required for the deep learning models are shown.

Dataset	Model	Epochs	RMSE	MAE	MAPE (%)	Time (sec)
Airline Passengers	ARIMA	-	47,318	39,065	9,5	0,3
	ES	-	54,328	44,673	10,9	0,1
	LSTM	500	36,763	26,777	6,6	9,9
	GRU	1000	41,832	30,979	7,3	15,0
Total COVID-19 cases	ARIMA	-	1309825,4	846478,9	0,1	2,1
	ES	-	1311301,1	844357,9	0,1	0,9
	LSTM	100	4655169,3	4251304,9	0,6	24,8
	GRU	100	7179840,4	6849942,7	1,1	24,9
Machine temperature	ARIMA	-	0,590	0,262	1,0	89,4
	ES	-	0,443	0,191	0,7	33,5
	LSTM	5	0,415	0,196	0,7	536,5
	GRU	50	0,405	0,218	0,8	653,4
SMHI temperature	ARIMA	-	1,288	0,909	-	594,6
	ES	-	0,839	0,574	-	252,2
	LSTM	5	0,660	0,437	-	2507,7
	GRU	5	0,658	0,436	-	2547,6

4.1.1 Airline Passengers

The deep learning models produced the most accurate predictions but needed many epochs for the optimal fit. Because of this, predicting with the deep learning models took a few seconds, whereas making a forecast using statistical models took just a fraction of that time. While ARIMA outperformed ES in terms of accuracy but not computing time, LSTM exceeded GRU in both areas.

4.1.2 Total COVID-19 Cases

The RMSE and MAE outcomes derived from the COVID-19 dataset are notably distinct from the other datasets. The scores exhibit considerable magnitudes, primarily attributed to the dataset's vast data points. Even so, the

grades are still relevant as they are not a consequence of errors, which the MAPE results confirm as it aligns with the other datasets.

GRU was an outlier in prediction accuracy for this dataset as the MAPE score is one per cent worse than the rest. Both statistical models had a similar prediction accuracy score, while LSTM outperformed GRU by the same margin as the statistical models outperformed LSTM.

The deep learning models required fewer epochs to fit the data optimally. As a result, the overfitting became a factor more rapid and was already a significant factor at 250 epochs. Overfitting refers to the model training too much on a dataset. Rather than capturing the underlying patterns or relationships, it starts to fit the noise or random fluctuations in the training data.

4.1.3 Machine Temperature

All models produced similar predictions for this dataset. The ARMA model performed slightly worse than all other models regarding prediction accuracy, but the statistical model's computing time was much faster than the deep learning models. GRU, LSTM and ES all produced very similar accuracy scores. However, LSTM and ES obtained a favourable MAE and MAPE grading, while the RMSE score was superior for the prediction given by the GRU model.

4.1.4 SMHI Temperature

As stated, the MAPE score produces disproportionately high scores when the observed data consists of values close to or equal to zero. As a result, the SMHI dataset's MAPE grading is useless because several of the temperature measurements contained therein equal zero.

The deep learning models produced the most accurate predictions for this dataset. However, the computing power and time necessary for the prediction were excessive. The GRU model outperformed LSTM in both time and accuracy on this dataset, but the difference in performance was minuscule.

The statistical models had trouble making accurate predictions; ARMA's RMSE and MAE scores were twice as high as deep learning models, while ES was slightly better. In contrast, the computing power and time required for the predictions were much lower than the deep learning models.

4.2 Multi-step forecasting

The results for the multi-step forecasting are shown in Table 3. Generally, multi-step forecasting differed in results from the single-step method. Statistical models achieved drastically worse outcomes in non-linear datasets

than deep learning models. Furthermore, SARIMA showed exponential growth in computing time with the size of the dataset.

Table 3: Result of the multi-step forecasting. Note that the total COVID-19 cases dataset was divided by ten million to retrieve forecasting from the deep learning models. Furthermore, the SARIMA model used minimised versions of both temperature datasets. * Applied on a reduced version of the dataset

Dataset	Model	Epochs	RMSE	MAE	MAPE (%)	Time (sec)
Airline Passengers	SARIMA	-	24,909	18,999	4,5	0,3
	ES	-	30,241	21,445	4,8	0,1
	LSTM	1900	77,128	58,220	12,8	32,3
	GRU	2700	74,574	55,979	12,2	59,5
Total COVID-19 cases	SARIMA	-	5,337	4,257	6,4	1,4
	ES	-	20,080	16,874	25,7	0,2
	LSTM	150	11,376	8,991	13,5	17,8
	GRU	350	13,4	10,953	16,4	67,5
Machine temperature	SARIMA*	-	4,696	3,435	9,8	16,2
	ES	-	4,365	3,571	14,8	0,1
	LSTM	7	0,423	0,232	0,9	29,4
	GRU	10	0,413	0,227	0,9	62,5
SMHI temperature	SARIMA*	-	4,390	3,476	-	72,3
	ES	-	8,258	6,780	-	1,9
	LSTM	2	0,707	0,480	-	38,9
	GRU	3	0,705	0,472	-	82,4

4.2.1 Airline passengers

The peaks in the airline passengers dataset were challenging for the deep learning models to predict. Both produced forecasts with a MAPE score of around 10%, where LSTM was faster but slightly less accurate than the GRU model. Similarly, for the statistical models, the ARIMA model was slightly more accurate than ES but also slower. Both statistical models had an improved accuracy grading compared to the deep learning models.

4.2.2 Total COVID-19 Cases

Both statistical models fitted the data to a straight line, whereas ARIMA performed best. Exponential smoothening got the best results when it assumed that the dataset did not have any seasonal component, conversely to ARIMA, which received the best score when it did account for a seasonal period. Both LSTM and GRU could not produce a prediction because of the immense values within the dataset. To solve this, the values within were divided by a factor of 10 000 000. Even then, the models failed to fit the data according to the constant linear trend that the dataset has as they approximated the prediction to a slightly lower value than the actual data.

4.2.3 Machine Temperature

Arima could not handle such large datasets, which resulted in the dataset being cut down to 8000 data points, where 5000 are training data. After that, it could produce a forecast which resulted in a straight line with a slight downward trend, but it still had a 10% error margin.

The computing time for the models differed immensely; ES produced a forecast of 0.3 seconds, and ARIMA took 16 seconds. LSTM and GRU took 37 and 54 seconds, respectively.

4.2.4 SMHI Temperature

Similarly to the machine temperature dataset, the dataset needed to be reduced to a smaller size for the ARIMA function to be able to produce a prediction. To keep the differences in size as a factor, the SMHI temperature dataset was reduced to 19 000 data points. In this case, 15 000 data points were used for training data and 4 000 for testing. ARIMA was very inefficient in its prediction. It took ARIMA longer to provide a forecast for the smaller dataset than for LSTM to produce a forecast for the complete dataset.

ES generated its prediction very fast. However, the accuracy of the ES model was the worst because it could not follow the local trends within the dataset and only oscillated around a specific value. Even so, the ES model was able to perform a prediction on the entire dataset as well. The prediction generated for the whole SMHI dataset was more accurate than the reduced version while still being worse than the deep learning models and ARIMA.

The deep learning models both generated very accurate forecasts. The predictions were equally precise for both versions of the dataset. In both cases, the RMSE and MAE score was better by a factor of seven to eight compared to the statistical models. Furthermore, the deep learning models required very few epochs to generate a good prediction, whereas GRU needed slightly more epochs to achieve comparable precision to LSTM.

5. Discussion

5.1 Summary

5.1.1 Single-step forecasting

No model outperforms the others in accuracy and computing time, but there are some evident patterns. The deep learning models performed very similarly to each other and produced more accurate results on all datasets than the statistical models except for total covid-19 cases. This dataset contained vast numerical values and a strong trend which appears to contribute to erroneous predictions. The effect of the immense numerical values is presumably an implementation-specific problem since values of considerable scale should not affect the accuracy of prediction models. It is also important to note that it took the deep learning models 22 seconds longer to produce a forecast than the second-best model, ARIMA, which took 1,4 seconds.

ES always had the lowest computing time and performed best on the total covid-19 cases and machine temperature dataset. However, the deep learning models were almost as accurate as ES on the machine temperature dataset. However, they took 9 and 11 minutes instead of ES, which needed 33,5 seconds to produce a forecast. It is further prominent in the SMHI dataset, where both deep learning models had a processing time of 40 minutes, and both statistical models were done in under 7 minutes. The rolling window implementation and the complexity of the deep learning models directly cause the extended computing time.

There is no evident connection between the characteristics of the datasets and the accuracy as was expected. As stated before, the rolling window eliminates all patterns of trend and seasonality because it forces the models only to observe three values at a time.

5.1.2 Multi-step forecasting

Multi-step forecasting favoured the deep learning models as the statistical models converged over time to a linear trend, except for the Airline passenger dataset. In contrast, exponential smoothening and ARIMA could accurately fit the data to the dataset. This is because the dataset has an evident seasonal pattern and trend and is partially linear.

The statistical models always had a lower computing time than the deep learning models, except for the SMHI data, where ARIMA took 72 seconds to forecast the smaller dataset version. While GRU, the next worse, took 81 seconds to predict the full version of the SMHI dataset accurately.

As previously mentioned, ARIMA could not perform multi-step forecasting on the larger datasets as it would run out of memory, which resulted in the datasets being reduced for that model. This causes the results to be more challenging to interpret as the model was based on a modified dataset compared to the other models. Furthermore, the computing time for the SMHI dataset is ten times larger than the machine temperature dataset while only consisting of three times more data points. However, this also causes the model to perform better on the dataset because it has more data to train on.

5.2 Correlations between characteristics and performance

5.2.1 Size

The size of the datasets mainly affects the computing time and power required for predictions. The deep learning models require fewer epochs to accurately predict large datasets, as opposed to small ones, where it needs more epochs to produce the same result. This directly affects the computing time as the models can run 1 000 epochs on the smaller datasets within the same time for a singular epoch on the large dataset. However, there is no visible correlation between the size of the datasets and the prediction accuracy.

5.2.2 Trend

The trend had a significant effect on the performance of the ARIMA model. ARIMA have greater prediction accuracy on all datasets containing a trend. According to our results, the ARIMA model is the best forecasting model for predicting datasets containing trends.

Contrastingly, the deep learning models failed to predict data containing trends accurately. Surprisingly, the deep learning models could only give a remotely accurate prediction for the total COVID-19 cases dataset if all data points were minimised in size. As previously mentioned, the size of the data points should not impact the prediction accuracy and is most likely implementation specific. Furthermore, only a minimal pre-processing was done on the data because this thesis concentrated on the forecasting models themselves. This might have led to the deep learning models failing to forecast trends accurately.

5.2.3 Seasonality

ES favours the datasets with a seasonal component and consequently performs worse on datasets without such characteristics. However, for the other models, the results provide no evidence of a correlation between the forecasts' accuracy and the dataset's seasonal component.

5.2.4 Linearity

The deep learning models were better at forecasting non-linear datasets than the statistical models.

Both statistical models were significantly affected by the linearity of the dataset. For multi-step forecasting, the statistical models cannot subsequently follow the non-linear data points and tend to create linear approximations for their predictions.

5.3 Limitations and future research

5.3.1 Limitations

One limitation of this study is the number of datasets. Due to the tight deadline of the thesis project, no more time could be allocated to finding and testing additional datasets. The limited number of different datasets made it challenging to find clear correlations between the data characteristics and the performance of the forecasting models.

The ARIMA model was not trained with optimal hyper-parameters. Therefore, the results may change with hyper-parameter-tuned ARIMA. This project used ACF and PACF plots as starting points. However, in most cases, trial and error were necessary because the plot reading was insufficient to identify the best model variant.

5.3.2 Future research

A potentially exciting extension of this project may be to investigate the parameter choices of the models. This means studying the effect of arbitrarily chosen parameters such as ARIMAs order, deep learning neurons and seasonality periods. Due to limited computing power or time, these parameters may be difficult to determine. Therefore, it can be interesting to see which models are less vulnerable to random parameters.

If the ARIMA model were hyper-parameter-tuned, it might have resulted in a higher order. This would, in turn, increase the rolling window size and could potentially lead to interesting results.

6. Conclusion

Deep learning models have demonstrated remarkable effectiveness in producing accurate results across various datasets. However, the efficacy of these models frequently relies upon the availability of large amounts of training data and a significant investment of time to optimise their performance. Despite their efficiency in single-step and multi-step forecasting, deep learning models may face limitations when applied to datasets with strong overarching trend components without sufficient pre-processing. This can pose a challenge for practitioners seeking to achieve optimal results with these models in domains characterised by high levels of trend variability. Therefore, it is essential to carefully consider the characteristics of the data and the problem at hand when evaluating the suitability of deep learning models for a particular task.

Statistical models have proven their ability to make precise forecasts in single-step forecasting. Nevertheless, statistical models typically fall short of deep learning models in performance regarding multi-step forecasting.

This is partly because statistical models demand much work to establish the definitive collection of parameters that can accurately forecast the specific dataset. On the other hand, deep learning models are more desirable since they can learn to detect relevant characteristics and patterns from the input data without needing assumptions or prior information about the underlying process.

7. References

[1] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[2] Z. Liu, Z. Zhu, J. Gao, and C. Xu, "Forecast methods for time series data: a survey," *Ieee Access*, vol. 9, pp. 91896-91912, 2021.

[3] G. U. Yule, "VII. On a method of investigating periodicities disturbed series, with special reference to Wolfer's sunspot numbers," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 226, no. 636-646, pp. 267-298, 1927.

[4] E. S. Gardner Jr, "Exponential smoothing: The state of the art—Part II," *International journal of forecasting*, vol. 22, no. 4, pp. 637-666, 2006.

[5] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.

[6] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[7] 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. 157-166, 1994.

[8] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, 2017: IEEE, pp. 1597-1600.

[9] J. H. F. Flores, P. M. Engel, and R. C. Pinto, "Autocorrelation and partial autocorrelation functions to improve neural networks models on univariate time series forecasting," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 2012: IEEE, pp. 1-8.

[10] E. Paparoditis and D. N. Politis, "The asymptotic size and power of the augmented Dickey–Fuller test for a unit root," *Econometric Reviews*, vol. 37, no. 9, pp. 955-973, 2018.

[11] D. G. Taslim and I. M. Murwantara, "A Comparative Study of ARIMA and LSTM in Forecasting Time Series Data," in *2022 9th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 2022: IEEE, pp. 231-235.

[12] P. T. Yamak, L. Yujian, and P. K. Gadosey, "A comparison between arima, lstm, and gru for time series forecasting," in *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, 2019, pp. 49-55.

[13] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, 2018: IEEE, pp. 1394-1401.

[14] Y. Aslam and N. Santhi, "A Review of deep learning approaches for image analysis," in *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2019: IEEE, pp. 709-714.

[15] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, 2010, vol. 57, no. 61: Austin, TX, pp. 10-25080.
[16] H. Ritchie *et al.*, "Coronavirus pandemic (COVID-19)," *Our world in data*, 2020.
[17] S. M. a. H. Institute. *Swedish meteorological data*,
[18] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, p. e623, 2021.
[19] M. V. Shcherbakov, A. Brebels, N. L. Shcherbakova, A. P. Tyukov, T. A. Janovsky, and V. A. e. Kamaev, "A survey of forecast error measures," *World applied sciences journal*, vol. 24, no. 24, pp. 171-176, 2013.