Rolling-Origin Evaluation Approach for Comparison of ARIMA and LSTM in Financial Time Series Forecasting

Mohamad Fasih

March 12, 2024

Abstract

Time series forecasting is a challenging task due to the stochastic nature of time series data. A variety of techniques have been utilized to forecast time series, including simple exponential smoothing, autoregressive (AR) moving average (MA), and integrated autoregressive moving average (ARIMA). Recently deep learning based model such as Long Short-Term Memory (LSTM) has been developed to forecast sequential data, particularly time series. In this paper, we compare the performance of two popular forecasting models, autoregressive integrated moving average (ARIMA) and long short-term memory (LSTM), on monthly data for the Wilshire 5000 Total Market Index from January 1990 to May 2023. The series is divided into training, validation, and test sets for finding optimal hyper-parameters and evaluating purposes. The rolling origin approach was applied to forecast the series values in the test set, and three evaluation criteria, including root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) employed to assess the model's performance. The results show that both models can be used to forecast total market indices, but LSTM was the superior model in this comparison in terms of all metrics. Additionally, ARIMA is less sensitive to changes in its hyper-parameters.

1 Introduction

Forecasting financial time series is important for making informed financial decisions. By forecasting the future values of financial variables such as stock prices, interest rates, and exchange rates, businesses and investors can make better decisions about how to allocate their resources. Time series analysis is a statistical technique that is used to analyze data that is collected over time such as financial time series. There are various objectives for studying time series including the understanding and description of the generating mechanisms, the forecasting of future values, and optimal control of a system. Time series by their very nature are dependent or correlated, so the order in which the observations are recorded is crucial. Hence, statistical procedures and techniques that rely on independence assumptions are no longer applicable, and different methods are needed [1]. Time series forecasting is a process to predict the future with the help of history data. Time series forecasting is based on the assumption that information will repeat in the near

future [2]. A common application of this type of data is tracking stock prices in finance. Time series analysis can be used to identify trends, patterns, and cycles in financial data. This information can then be used to make predictions about future prices. Forecasting the future values of an observed time series plays an important role in nearly all fields of science and engineering, such as finance, business intelligence, meteorology, and telecommunications [3]. Good forecasts are also vital in many other areas of scientific, industrial, commercial, and economic activity [4]. Since stock market prediction is regarded as a challenging task of financial time-series prediction [5], the finance industry has always been interested in the successful prediction of financial time series data [6]. The methods used for forecasting can also vary widely. Some forecasting methods are simple, such as the naive method, which uses the most recent observation as a forecast. Other forecasting methods are more complex, such as neural nets and econometric systems of simultaneous equations [7]. Time series forecasting is a common problem and several approaches have been used to address this problem. Box and Jenkins developed integrated autoregressive moving average (ARIMA) methodology to fit a class of linear time series models [8]. The great advantage of ARIMA models is that they have the flexibility and ability to handle a variety of time series data. These models are considered one of the most extensively used in the economics and finance fields [9]. The reasons are that ARIMA models are easy to implement and interpret, as they only require a few parameters and can also provide reliable forecasts and confidence intervals, as they are based on statistical methods and theory [10]. ARIMA models can account for some patterns, such as linear trends and seasonal fluctuations. [11] used the seasonal integrated autoregressive moving average model (SARIMA) which is formed by including additional seasonal terms in the ARIMA to forecast the exchange rate of the Jordanian Dinar and the US Dollar and compared it with the ARIMA model. The SARIMA model outperformed the ARIMA in terms of four accuracy metrics including RMSE, MAE, MAPE, and MASE. The forecasting domain has been influenced, for a long time, by linear statistical methods such as ARIMA models. ARIMA models are a popular choice for modeling economic and financial time series; however, they have some limitations. For example, the ARIMA model has a basic assumption that the variance of the error terms is constant however, this assumption may not be met in reality. Implementing Generalized Autoregressive Conditional Heteroscedasticity (GARCH) could tackle this obstacle. In GARCH models, it is assumed that the variance of the error term follows an autoregressive moving average process [12]. Another drawback of ARIMA models is that simple ARIMA models cannot model non-linear relationships between variables. This can be a problem if the relationships between the variables in the time series are non-linear. It has been shown in [13, 14] that linear models are not well suited to many practical applications. The results of [15] and [16] showed that, due to the high complexity of financial time series, simple linear ARIMA based modeling techniques may not adequately model financial data. To overcome the linear limitation in time series models, several non-linear models have been proposed. including the bilinear model [17], the threshold autoregressive (TAR) model [18], and chaotic dynamics [19]. They may work well for a particular situation, but they are not suitable for all situations. Due to their focus on specific non-linear patterns, these models cannot model other types of non-linearity. More recently, Artificial Neural Networks (ANN) have been studied and may be a suitable modeling technique for these types of data [20–22]. For example, To address the non-linear behavior of exchange rate time series, [23] employed two prominent neural network architectures, Multilayer Perceptron (MLP) and Radial Basis Function (RBF), to effectively capture the non-linear patterns

that eluded the ARIMA model. Since in the last two decades, machine learning models have drawn attention and have established themselves as serious contenders to classical statistical models in the forecasting community, different approaches have been developed in this area of research. Long Short Term-Memory (LSTM)is a common deep learning based model used for time series forecasting [24]. [25] introduced the CNN-LSTM model, a combination of convolutional neural network (CNN) and Long Short-Term Memory to use the advantage of both models. While neural networks like LSTM offer significant advantages in time series forecasting, they also present certain drawbacks, such as substantial data requirements and computational expensiveness. Both ARIMA and LSTM can be effectively utilized to predict stock market price [26, 27]. All these forecasting methods can be assembled with rolling origin forecasting to evaluate the performance of these models and also compare them to each other. This technique considers moving forward the training set and updating it with new values from the test set. For instance, [28] exploited this strategy to assess the forecasting accuracy of the ARIMA model. In that study, different ARIMA models were compared to each other by focusing on different periods of the forecast. Similar research can be found in [29] by additional consideration to either re-estimating or not re-estimating the parameters of the model after each updating the training set to compare the different smoothing methods. The rolling origin evaluation technique holds a variety of approaches because it uses different values for forecasting the horizon and the number of steps to forecast the desired horizon. There is a comprehensive work around the variation on rolling origin evaluation in [30]. [31] employed the rolling origin evaluation technique to compare the forecasting accuracy of ARIMA and LSTM in both univariate and multivariate situations for hand, foot, and mouth disease (HFMD) data and found that both univariate and multivariate LSTM had more precise forecasts than ARIMA. For the financial and economic time series data, research by [32] showed that the LSTM outperformed ARIMA for twelve datasets including six different monthly economic datasets and six different monthly financial datasets. In this paper, we compare the performance of the LSTM model with the ARIMA model in the forecasting of the Wilshire 5000 Total Market Index which is a market-capitalization-weighted index of the market value of all American stocks. A rolling forecast evaluation technique was applied to compare both models. The main contributions of this paper are: to conduct a rolling origin forecasting evaluation for ARIMA and LSTM and also compare the execution of two models with respect to three error measurement tools including RMSE, MAE, and MAPE. The outline of the chapter is as follows. Section 2 provides some basic theoretical notions of ARIMA and LSTM models. Section 3 presents the results of the study, followed by the discussion about the findings and results in section 4. Finally, section 5 concludes the paper.

2 Methods

2.1 ARIMA

Autoregressive integrated moving average is a general class of models for stationary time series. It is an extended form of the autoregressive moving average (ARMA) model which is a combination of autoregressive (AR) and moving average (MA). ARIMA models are integrated type of ARMA models and can be understood by outlining each of its components as follows:

AR(P): Autoregressive processes are as their name suggests regressions on themselves. Specifically, a pth-order autoregressive process X_t satisfies the equation:

$$\phi(B)X_t = Z_t \tag{1}$$

where $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$ is known as autoregressive polynomial and Z_t is a sequence of independent , identically distributed random variables (usually from normal distribution) called white noise with mean zero and variance σ^2 or briefly, $Z_t \sim IID(0,\ \sigma^2)$ B is the backshift operator, that is, $B^k X_t = X_t - X_{t-k}$. Without using the backshift operator equation (1) can be written as $X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + Z_t$.
MA(q): A moving-average model is conceptually a linear regression of the current value of the series against current and previous (observed) white noise error terms. The equation of the moving average of order q can be written as follows:

$$X_t = \theta(B)Z_t \tag{2}$$

where $\theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q$ is known as autoregressive polynomial and Z_t and B are white noise and backshift operator respectively. The alternative representation of (2) is $X_t = \sum_{i=1}^{q} \theta_i Z_{t-i} + Z_t$ and θ's are the parameters of the model.
ARMA(p, q): If we assume that the series is partly autoregressive and partly moving average, we obtain a quite general time series model called ARMA which can be written as follows:

$$\phi(B)X_t = \theta(B)Z_t \tag{3}$$

The equation (3) can also be represented $X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{i=1}^{q} \theta_i Z_{t-i} + Z_t$ which makes it look more tangible.
ARIMA(p, d, q): Autoregressive integrated moving average (ARIMA) is a generalization of ARMA models. That is the integrated form of ARMA models. Y_t is an ARIMA(p, d, q) process, if the following process will be an ARMA(p, q) process:

$$X_t = Y_t - Y_{t-d} \tag{4}$$

ARIMA models were developed in 1930 by George Box and Gwilym Jenkins and as a result, ARIMA models are also known as Box-Jenkins models. These models are also very suitable for non-stationary time series because of their differencing features. The goal of differencing is to remove any trends or seasonality from the data in order to make it stationary, which is required for ARIMA models to produce accurate forecasts. ARIMA models are typically described by a three-tuple (p, d, q). The first number, p, represents the order of the autoregressive (AR) component of the model. The second number, d, represents the number of times the time series has been differenced. The third number, q, represents the order of the moving average (MA) component of the model. The first step in the procedure is to difference the time series until it is stationary. This means that the time series has no trend or seasonality. In many cases, one or two stages of differencing are sufficient. The differenced series will be shorter than the original series by the number of times it has been differenced. Once the time series is stationary, the ARMA model is fitted to it. The values of p, d, and q are chosen by trial and error. There are a few principles that can be used to guide the choice of p, d, and q. First, the model should be as simple as possible. This means that the values of p and q should be as small as possible. Second, the model should fit the historical data as well as possible. Third, the partial autocorrelation at lags 1, 2, 3,... should provide an indication of the order of the

AR component, i.e. the value chosen for q. Finally, the shape of the autocorrelation function (ACF) plot can suggest the type of ARIMA model required. It is also possible to include seasonal components in ARIMA models. These models are more complex to fit and interpret, but they can be more accurate for forecasting seasonal data. The seasonal components are also described by a three-tuple (P.D.Q) where P is the seasonal AR order, D is the seasonal differencing, and Q is the seasonal MA order.

2.2 LSTM

Neural networks (NN) are a type of machine learning algorithm that have a wide range of applications in many fields. They are made up of interconnected nodes, or neurons, that learn to process information and find patterns from the data.
ANN: Artificial neural networks ANN are typically composed of three layers including an input layer, one or more hidden layers, and an output layer. The input layer receives the data that the ANN will process. The hidden layers perform the actual processing of the data. The number of hidden layers and the number of neurons in each layer can vary. The output layer produces the results of the processing. The connections between the neurons are weighted. These weights are adjusted during the training process. The training process for ANNs is typically done using a supervised learning algorithm. This means that the ANN is given a set of input data and the desired output data. The ANN then tries to learn the relationship between the input data and the output data. Once the ANN is trained, it can be used to process new data. The ANN will output a prediction for the new data. Neural networks can be used to solve a wide variety of problems, including classification, regression, and prediction.
RNN: Recurrent neural networks (RNNs) are a type of neural network that are specifically designed to handle sequential data such as time series. RNNs work by maintaining a hidden state that stores information about the previous inputs. This allows them to learn long-term dependencies in the data, which is essential for tasks such as machine translation and speech recognition.
LSTM: Long short-term memory (LSTM) networks are a special type of RNN that were developed to address some limitations of RNN. When it comes to long-term data-dependent problems, RNNs are hardly competent. LSTMs have a special structure that allows them to learn long-term dependencies and remember information that was presented to them many time steps ago in the data [33]. This makes them well-suited for tasks such as time series forecasting. They have been used to achieve state-of-the-art results on a variety of tasks, including machine translation, speech recognition, and natural language processing [34–36]. There are additional reasons that LSTM models are suitable tools to use for time series data such as their ability to learn complex patterns in the data and handling non-linearity which is common in time series data. In addition, they are useful for noisy data which is also common in time series frameworks. LSTMs have become one of the most popular types of RNNs. LSTM neurons can remember information from the past, update it, and pass it on to the next layer or cell without losing any information. These types of neural networks use three gates to control information flow: input gate, forget gate, and output gate. The input gate controls how much new information from the current time step is added to the LSTM cell state. The forget gate controls how much information from the previous time step is forgotten. The output gate controls how much information from the LSTM cell state is output to the next layer of the network. Figure 1 illustrates this process.

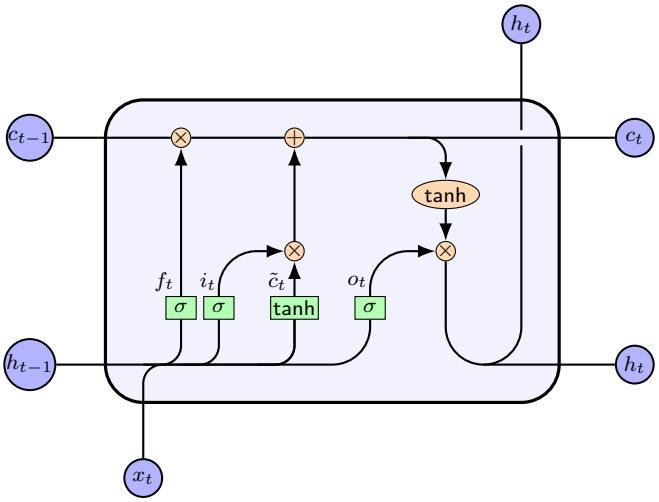

Figure 1: The structure of Long Short-Term Memory (LSTM) cell.

Forget gate: It determines whether information in the middle and previous layers should be retained or discarded. The forgetting gate function can be expressed as follows:

$$f_t = \sigma\left(w_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{5}$$

Input gate: The input gate is followed by the forgetting gate, which updates the data and combines it with the storage unit. The following is the specific formula:

$$i_t = \sigma\left(w_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{6}$$

Output gate: The output gate determines the model output based on the weight of the control state Ct. The activation function obtains the initial output, which is then normalized by the tanh function. The expression is as follows:

$$o_t = \sigma\left(w_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{7}$$

$$h_t = o_t \odot \tanh c_t \tag{8}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{9}$$

$$\tilde{c}_t = \tanh\left(w_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{10}$$

In the above equations, w_f, w_i, w_o, w_c denote the weight of the forgetting gate, input gate, output gate, and memory cells respectively. Also b_f, b_i, b_o, b_c present the bias value for of the forgetting gate, input gate, output gate, and memory cells respectively. All the weights and biases are generated by random initialization. σ is the sigmoid activation function and the operator $\odot$ represents the element wise multiplication of the vectors.

2.3 Rolling-origin evaluation

It is important to evaluate the accuracy of forecasts using real-world data. Residuals (the difference between the actual and predicted values) can be used to assess the adequacy of a model by checking their distribution and plot. A good model will have residuals that are:

- Randomness: The residuals should not show any systematic patterns; such as trends or seasonality.
- Homoscedastic: The residuals should have constant variance. This means that the spread of the residuals should be the same for all values of the fitted values.
- Normally distributed: The residuals should be normally distributed. This means that they should follow a bell-shaped curve.

If the residuals do not meet these criteria, then the model may not be adequate. For example, if the residuals are not random, then this may indicate that the model is not capturing all of the variability in the data. If the residuals are not homoscedastic, then this may indicate that the model is not capturing the changes in variance over time. If the residuals are not normally distributed, then this may indicate that the model is not capturing the underlying distribution of the data. However, this assessment criterion may not be useful when it comes to evaluating the accuracy of forecasting. The size of the residuals is not a reliable indicator of how accurate future forecasts will be. There are some notes that should be considered about the importance of the accuracy of forecasting models. A model that fits the training data well does not necessarily mean that it will forecast well. Methods selected by best in-sample fit may not best predict post-sample data [37]. n the other hand, overfitting and structural changes may further aggravate the divergence between in-sample and post-sample performance. A perfect fit can always be obtained by using a model with enough parameters. However, this does not mean that the model is accurate. In fact, a model with too many parameters may be overfitting the data, which means that it is memorizing the noise in the data instead of learning the underlying patterns. This is because the training data may contain noise or outliers that are not representative of the true underlying distribution of the data. Overfitting a model to data is just as bad as failing to identify a systematic pattern in the data. This is because an overfitted model will not be able to generalize to new data. The only way to truly assess the accuracy of a forecast is to see how well it performs on data that it was not trained on. When choosing forecasting models, it is common practice to split the available data into two sets: training data and test data. The training data is used to estimate the parameters of the forecasting model, while the test data is used to evaluate the accuracy of the model. The test data is not used in determining the forecasts, so it provides a reliable indication of how well the model will perform on new data. The size of the test set is typically about 20% or 30% of the total sample, although this value can vary depending on the length of the sample and the forecasting horizon. The test set should ideally be at least as large as the maximum forecast horizon required. A common technique that is widely used to measure and evaluate the accuracy of forecasting models is rolling-origin evaluation also called walk-forward validation. To begin, the model is optimized with in-sample data, and the remaining data is reserved for out-of-sample testing. A small portion of the reserved data following the in-sample data is tested and the results are recorded. The in-sample time window is shifted forward by the out-of-sample period, and the process is repeated. Finally, the model's performance is evaluated based on all recorded results. This evaluation technique can be used for both model selection and comparison of different models. Rolling-origin forecasting is a form of out-of-sample forecasting, which means that the model is not trained on the data that it is being used to forecast. This is in contrast to in-sample forecasting, where the model is trained on the entire dataset, including the data that it is being used to forecast. For

out-of-sample evaluations of forecasting accuracy, historical data series are divided into a fit period (training set) and a test period. The final time in the fit period - the point from which the forecasts are generated is the forecasting origin. The number of time periods between the origin and the time being forecast is the lead time or the forecasting horizon. Another notable problem here is the re-estimation (re-optimization) of series in every updated forecasting origin. After Updating the origin, model parameters can be either estimated again or remain as they were in the previous step. Updating origins without re-estimating the model's parameters prevents the forecasting method from reaching its full potential and re-estimating the model's parameters after every updating ends with a lower forecasting error [30]. A comprehensive research by [29] showed these results by checking different leads. Nevertheless, rolling origin with re-estimation can be computationally more intensive than simply updating without re-estimating. Rolling origin forecasting is a more realistic approach to forecasting than in-sample forecasting because it allows the model to learn from the most recent data. However, it can also be more computationally expensive, because the model needs to be re-fitted for each time step. There are three types of rolling forecasting which are commonly used [38]:

- One-step forecasts without re-estimation: One-step forecasts are the simplest type of rolling forecast. On a single set of training data, the model is estimated, and then one-step forecasts are computed on the remaining test data.
- Multi-step forecasts without re-estimation: Multi-step forecasts are more complex than one-step forecasts. In this approach, the model is fitted on the training data and used for forecasting multiple time steps ahead of the test data. This type of forecast does not require the model to be re-estimated after each forecast.
- Multi-step forecasts with re-estimation: Multi-step forecasts with re-estimation are the most complex type of rolling forecasts. The training data is extended in every iteration and refitted before each forecast is computed. This type of forecast allows the model to learn from the most recent data, which can improve the accuracy of the forecasts.

2.4 Data Analysis

2.4.1 Dataset

With the goal of analyzing and modeling, we selected monthly time series of the Wilshire 5000 Total Market Index, or more simply the Wilshire 5000 from the Federal Reserve Bank of St. [1] The Wilshire 5000 is a stock market index that tracks the performance of the 5,000 largest publicly traded companies in the United States and therefore, it is a valuable tool for investors who want to track the performance of the overall US stock market. The monthly data set covers the period from January 1990 to May 2023.

2.4.2 Assessment Metrics

Three accuracy metrics were calculated to evaluate the ARIMA and LSTM model's performance on the test set including root mean square error (RMSE), mean absolute per-

[1] https://fred.stlouisfed.org/.

centage error (MAPE), and mean absolute error (MAE). These three accuracy metrics are computing as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(X_i - \hat{X}_i\right)^2} \tag{11}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|X_i - \hat{X}_i\right| \tag{12}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|X_i - \hat{X}_i\right|}{X_i} \times 100 \tag{13}$$

In the above equations, X_i is the monthly index of Wilshire 500, and $\hat{X}_i$ is its predicted value. In RMSE, since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the RMSE is most useful when large errors are particularly undesirable. As a matter of fact, it penalizes large errors. Unlike the RAMSE in MAE, all the individual errors are weighted equally in the average and because it uses the absolute difference between actual and predicted values, it's not sensitive to outliers. MAPE is the percentage form of MAE, which makes it more understandable and interpretable than MAE.

2.4.3 Data splitting

The series observations from January 1990 to December 2018 were selected as the training set, and the series observations from January 2019 to May 2023 were selected as the test set. Usually, a model is constructed on the training set and evaluated on the test set, but there is always a desire to have a more robust technique to reach a more precise model. This purpose can be achieved when the model's performance is evaluated several times. So we can have more subsets of dataset to investigate the performance of the model multiple times. We divided the training set into four pairs of sub-training and validation sets. The first sub-training set is from January 1990 to December 2014, and the corresponding validation set is from January 2015 to December 2015. The second sub-training set is from January 1990 to December 2015, and the corresponding validation set is from January 2016 to December 2016. The next two pairs of sub-training and validation sets were accomplished in the same way.

2.4.4 Model Selection

For the both ARIMA and LSTM models we fit different models on every sub-training set and predicted its matching validation set then calculated the RMSE. The model that had the lowest average RMSE on all four validation sets, was selected as the predictive model. To select a model, we followed a process which is called hyper-parameters tuning in machine learning context. In a model, a hyper-parameter refers to a characteristic that cannot be estimated based on data and is external to the model. Before beginning the learning process, the hyper-parameter value must be set. A powerful tool that is vastly used to achieve this goal is grid search. The grid-search algorithm is used to determine the model's optimal hyper-parameters so that the predictions can be made with the greatest degree of accuracy. For the ARIMA model, the order of the autoregressive part p, and the

moving average part q are the hyper-parameters that have to be set and for the LSTM model, the number of time steps and the LSTM units were considered to be the model's hyper-parameters. All the models regarding to these hyper-parameters were fitted to the first sub-training set and predicted the corresponding validation set then the forecasting origin was moved forward by 12 units and all the models were built on the updated sub-training set and predicted the next validation set. This is a 12-step ahead rolling origin forecasting used for finding the optimum model for both ARIMA and LSTM. The model with the lowest average RMSE on validation sets was selected as a candidate model to use for the comparison procedure of ARIMA and LSTM.

2.4.5 Comparison of ARIMA and LSTM

To attain the main objective of this study each ARIMA and LSTM selected model by the structure explained in the previous section, was built on the training set which starts from January 1990 and ends with the last element of the last validation set and then used to predict the test data and compared the forecasting performance of two methods. To accomplish this aim, rolling-origin evaluation can be used again, but this time we predicted a single value in each step. More clearly the first value of the test set was predicted and its actual value was added to the train set, after re-estimating the parameters, the next value was predicted. This approach mimics the real-world trend and gives remarkable feedback on model performance. All the mentioned assessment metrics RMSE, MAE, and MAPE were calculated for both ARIMA and LSTM methods to compare them to each other.

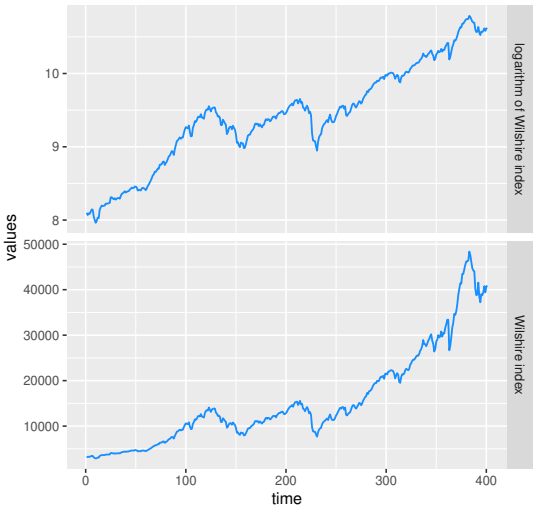

Figure 2: Wilshire 5000 index from january 1990 to May 2023.

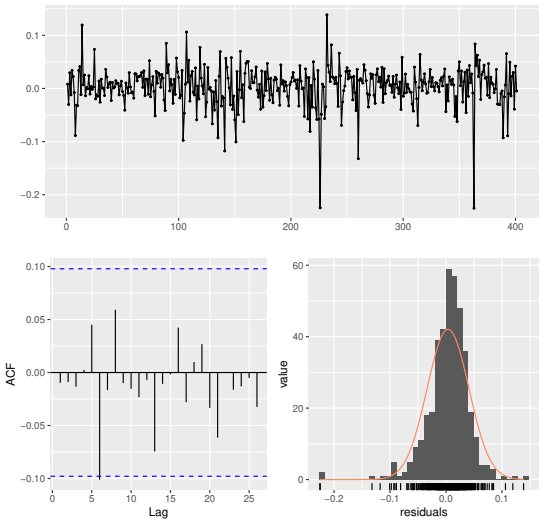

Figure 3: Residuals analysis of the ARIMA(1, 1, 4) model.

3 Results

3.1 ARIMA

First of all, a stationary dataset is a prerequisite for employing ARIMA models. The dataset appears to have an exponential trend at first glance. This is shown in the bottom plot of Figure 2. The augmented Dicky-Fuller test also showed the non-stationarity of time series with the P-value of 0.731. The top plot of Figure 2 depicts the transformed time series using the logarithmic transformation which is a common way to transform non-stationary time series with exponential trends. Since the data continue to show an increasing linear trend, it requires the first-order differential transform. Grid search was used for ARIMA parameters to find the best model that has the minimum average value for RMSE on four validation sets. Values for p and q were set from 0 to 4. We fitted all the models to sub-training data and predicted corresponding validation. For All models, the average value of the root mean squared error (RMSE) of validation sets is presented in Table 1. According to Table 1, ARIMA (1, 1, 4) had the lowest average RMSE on validation sets which means that it performed well on all validation sets generally. Therefore, the ARIMA (1, 1, 4) was selected as the best model between ARIMA models. This model is utilized for comparison with the chosen LSTM model on the test set.

3.2 Checking the Adeccuacy of ARIMA

The adequacy of the model was investigated by checking for autocorrelation in the residuals. Figure 3 shows the plots associated with the residuals. The time series plot of residuals does not show any signs of a clear pattern. The autocorrelation plot illustrates that there is no evidence of autocorrelation between residuals. The Ljung-Box test was also employed to ensure that the model met the assumption of the withe noise for the residuals. Since the P-value of the Ljung-Box test was 0.891 it is accepted to consider residuals uncorrelated.

3.3 LSTM

We used the number of LSTM units and the number of lags as hyperparameters and set six different values for the number of lags and five different values for the LSTM. All the models were trained on all four sub-training sets and evaluated on corresponding validation sets using root mean squared error. Table 2 presents the findings. As indicated by Table 2, the model with 15 lags and 64 units had the lowest average RMSE on validation sets. Hence, we selected this model as the predictive model to build on the training set and forecast the test set values.

3.4 Checking the Overfitting of LSTM

Overfitting occurs when a model performs well on training data but poorly on test data. In this situation, the test error is much higher than the training error. To explore whether the selected LSTM model shows any behavior that implies overfitting or not, we used the learning curve plot which is a widely used diagnostic tool in machine learning for algorithms that learn from a training dataset. As shown in Figure 4, the plot does not

Table 1: RMSE for different ARIMA models on validation sets

Model	p	q	Validation RMSE
1	1	0	1224.56
2	2	0	1246.23
3	3	0	1201.23
4	4	0	1174.83
5	0	1	1257.48
6	1	1	1251.52
7	2	1	1238.99
8	3	1	1156.17
9	4	1	1168.81
10	0	2	1253.67
11	1	2	1149.12
12	2	2	1145.77
13	3	2	1144.94
14	4	2	1222.23
15	0	3	1231.91
16	1	3	1150.33
17	2	3	1144.82
18	3	3	1209.12
19	4	3	1224.61
20	0	4	1205.67
21	**1**	**4**	**1119.50**
22	2	4	1241.92
23	3	4	1194.32
24	4	4	1137.00

Table 2: RMSE for different LSTM models on validation sets

Model	Lags	LSTM units	Validation RMSE
1	2	4	1816.02
2	2	8	1313.25
3	2	16	1287.83
4	2	32	1106.57
5	2	64	988.14
6	3	4	1511.91
7	3	8	1402.70
8	3	16	1473.90
9	3	32	1234.51
10	3	64	984.08
11	4	4	1693.38
12	4	8	1329.18
13	4	16	1127.67
14	4	32	1112.53
15	4	64	1063.30
16	5	4	1506.73
17	5	8	1439.30
18	5	16	1502.63
19	5	32	1146.78
20	5	64	1116.19
21	10	4	1913.27
22	10	8	1490.72
23	10	16	1370.79
24	10	32	1360.81
25	10	64	1244.77
26	15	4	1365.82
27	15	8	1238.12
28	15	16	1296.06
29	15	32	1226.21
30	**15**	**64**	**937.17**

show any sign of overfitting since the loss values of the training and test sets decreased by the number of iterations.

3.5 Performance on validation sets

We have used validation sets to choose the best model for both ARIMA and LSTM and compare them on the test set. However, it is useful to check the overall performance of models on validation sets. The distribution of average root mean squared error for all ARIMA and LSTM models was investigated. The summary statistics of all ARIMA and LSTM models are presented in Table 3. The mean and standard deviation of the average RMSE for the ARIMA models is lower than Those of LSTM. This can implicate that ARIMA is less sensitive to its hyper-parameters than LSTM. While the minimum average of RMSE for the LSTM is lower than the minimum average of RMSE for the ARIMA. Figure 5 shows the distribution of average RMSE for ARIMA and LSTM models using boxplot.

3.6 Comparison of ARIMA and LSTM

In the final stage, we investigated the performance of selected ARIMA and LSTM models on the test set using rolling origin forecast with forecasting a single value in the test set in each step. Table 4 displays the values of RMSE, MAE, and MAPE for the ARIMA(1, 1, 4) and the LSTM with 64 units and 15 time steps. According to TABLE IV, both models performed well on the test set. However, LSTM outperformed ARIMA in the prospective forecasting of the Wilshire 5000 total market index from January 2019 to May 2023, with the smaller values of RMSE (1192.64), MAE (931.75), and MAPE (2.52). Figure 6 visualizes the predicted and actual values of the ARIMA and LSTM models.

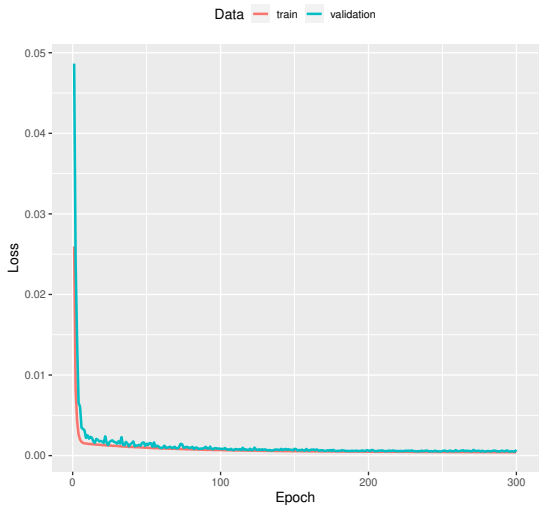

Figure 4: Validation loss versus training loss for the LSTM.

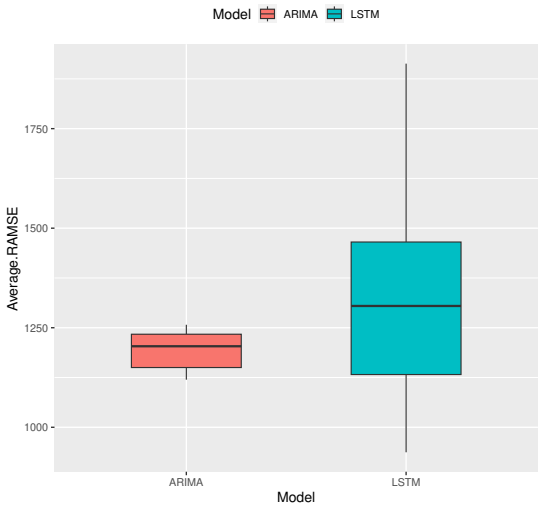

Figure 5: Distribution of RMSE for ARIMA and LSTM.

Table 3: Descriptive statistics of error for ARIMA and LSTM

Model	Statistics			
	Mean	Standard Deviation	Min	Max
ARIMA	1195.62	44.09	1119.50	1257.48
LSTM	1320.01	233.81	937.17	1913.27

Table 4: The forecasting performance of two models according to three metrics.

Model	Accuracy Metric		
	RMSE	MAE	MAPE
ARIMA	1677.162	1230.42	3.48
LSTM	1192.642	931.75	2.52

4 Discussion

In this study, a comparative analysis was done to investigate the forecasting performance of two frequently used models - a traditional statistical model ARIMA and a deep learning based model LSTM. Each model has some advantages and disadvantages. LSTM is a more complex model than ARIMA and it can catch more complexity such as nonlinearity from the data. Additionally, LSTM uses long-term dependencies to forecast future values. While ARIMA is more suitable for data that include linearity and yields interpretable results. [32, 39]. On the other hand, the optimization of the neural network model is a very complex technical challenge. An LSTM model is susceptible to overfitting and it is required to tune numerous hyper-parameters which makes it more time-consuming than the ARIMA [33]. In our study, different values were set for the order of autoregressive and moving average parts for ARIMA. The number of LSTM units and the number of lags were used as hyper-parameters to tune for LSTM. The average RMSE of all different ARIMA and LSTM models on validation sets was used to compare the overall performance of the two methods. Results illustrated That ARIMA is more robust to changes in its hyper-parameters since the Average RMSE of all ARIMA models on validation sets was more concentrated over its mean compared to the LSTM. In addition, the average RMSE of ARIMA models on validation sets had a lower mean compared to LSTM, however, the difference between the mean of both methods was not high and both methods performed well on validation sets. In terms of performance on the test set, rolling origin with one step ahead forecast, was implemented because it seems it is more consistent with the incremental nature of real change of the series over time. Three metric assessments including RMSE, MAE, and MAPE were used to measure the accuracy of both models. However, both ARIMA and LSTM had an acceptable performance on the test set, LSTM outperformed ARIMA according to all three metrics. Prior studies also found LSTM to be more accurate than ARIMA. [31, 32].

5 Conclusion

In this study, we compared two popular models in the forecast of the monthly index for the Wilshire 5000 Total Market from January 1990 to May 2023. We selected the ARIMA

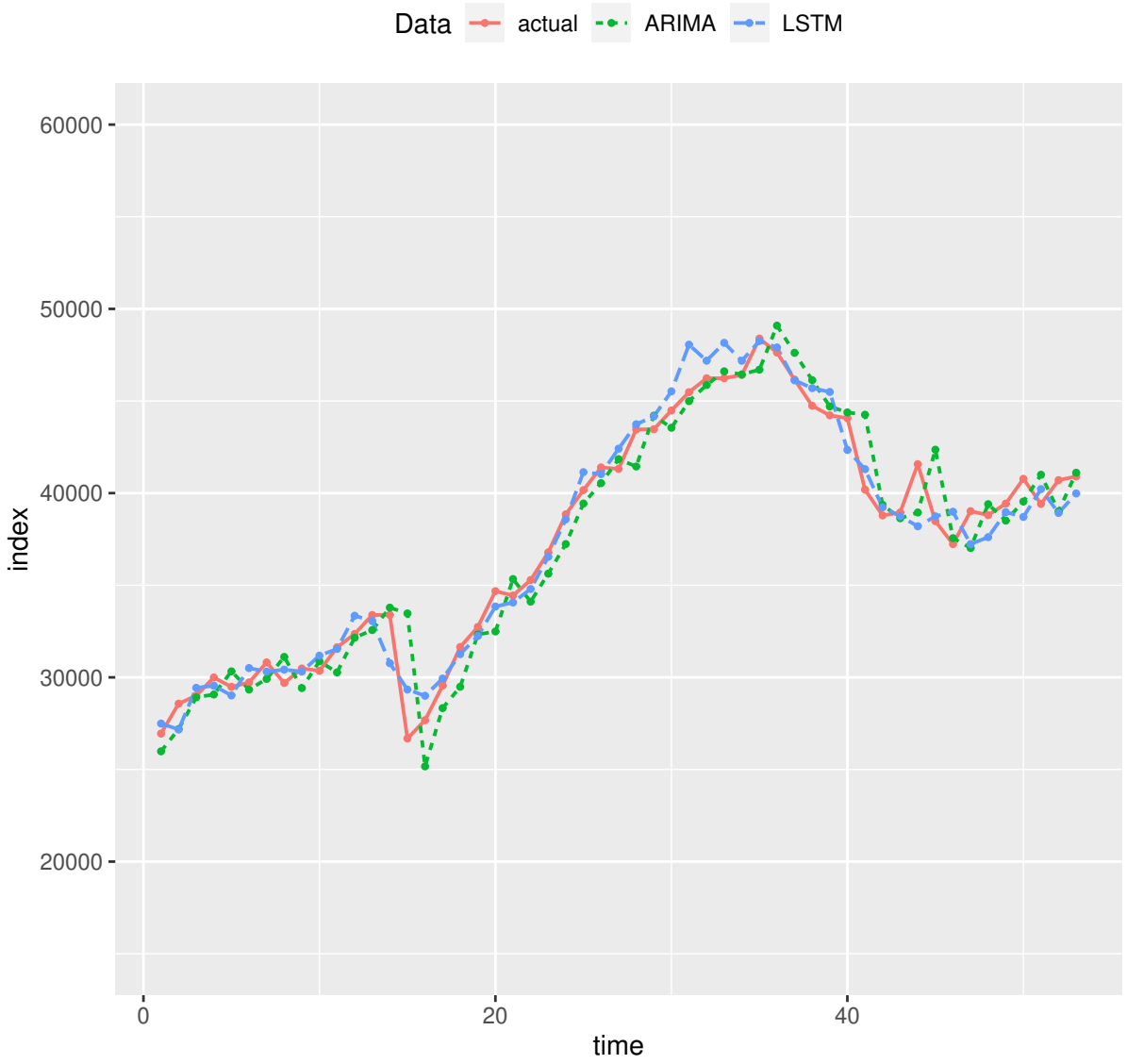

Figure 6: Comparison between actual and prediction for the Wilshire 5000.

and LSTM models among the models with different hyper-parameters with the lowest RMSE and used them for prediction purposes. Our findings showed that both models demonstrated strong forecasting performance however LSTM produced more accurate forecasts than ARIMA.

References

[1] W. W. S. Wei, Time Series Analysis: Univariate and Multivariate Methods. Pearson Addison Wesley, 2006.

[2] S. Athiyarath, M. Paul, and S. Krishnaswamy, "A Comparative Study and Analysis of Time Series Forecasting Techniques," 2020.

[3] G. Bontempi, S. B. Taieb, and Y.-A. L. Borgne, "Machine Learning Strategies for Time Series Forecasting".

[4] C. Chatfield, Time-series forecasting. Boca Raton, Fla.: Chapman & Hall/CRC, 2000.

[5] K. Kim, "Financial time series forecasting using support vector machines," Neurocomputing, vol. 55, no. 1, pp. 307–319, Sep. 2003, doi: 10.1016/S0925-2312(03)00372-2.

[6] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," Appl. Soft Comput., vol. 90, p. 106181, May 2020, doi: 10.1016/j.asoc.2020.106181.

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

[8] G. Box, "Box and Jenkins: Time Series Analysis, Forecasting and Control," in A Very British Affair, London: Palgrave Macmillan UK, 2013, pp. 161–215. doi: 10.1057/9781137291264_6.

[9] S. Khan and H. Alghulaiakh, "ARIMA Model for Accurate Time Series Stocks Forecasting," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 7, 2020, doi: 10.14569/IJACSA.2020.0110765.

[10] [1] G. Perone, An Arima Model to Forecast the Spread and the final size of COVID-2019 Epidemic in Italy (first version on SSRN 31 March). 2020. doi: 10.2139/ssrn.3564865.

[11] [1] R. Al-Gounmeein and M. T. Ismail, "Forecasting the Exchange Rate of the Jordanian Dinar versus the US Dollar Using a Box-Jenkins Seasonal ARIMA Model," Jan. 2020.

[12] [1] C. Francq and J.-M. Zakoian, GARCH Models: Structure, Statistical Inference and Financial Applications. John Wiley & Sons, 2019.

[13] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," Neurocomputing, vol. 50, pp. 159–175, Jan. 2003, doi: 10.1016/S0925-2312(01)00702-0.

[14] P.-F. Pai and C.-S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," Omega, vol. 33, no. 6, pp. 497–505, Dec. 2005, doi: 10.1016/j.omega.2004.07.024.

[15] P. Escudero, P. Paredes-Fierro, and W. Alcocer, "Recurrent Neural Networks and ARIMA Models for Euro/Dollar Exchange Rate Forecasting," Appl. Sci., vol. 11, p. 5658, Jun. 2021, doi: 10.3390/app11125658.

[16] M. Khashei, M. Bijari, and G. A. Raissi Ardali, "Improvement of Auto-Regressive Integrated Moving Average models using Fuzzy logic and Artificial Neural Networks (ANNs)," Neurocomputing, vol. 72, no. 4, pp. 956–967, Jan. 2009, doi: 10.1016/j.neucom.2008.04.017.

[17] T. S. Rao and M. M. Gabr, An Introduction to Bispectral Analysis and Bilinear Time Series Models. Springer Science & Business Media, 2012.

[18] C. W. S. Chen, F.-C. Liu, and M. K. P. So, "A review of threshold time series models in finance," Stat. Interface, vol. 4, no. 2, pp. 167–181, 2011, doi: 10.4310/SII.2011.v4.n2.a12.

[19] M. Vogl, "Controversy in financial chaos research and nonlinear dynamics: A short literature review," Chaos Solitons Fractals, vol. 162, p. 112444, Sep. 2022, doi: 10.1016/j.chaos.2022.112444.

[20] S. Mehtab and J. Sen, Analysis and Forecasting of Financial Time Series Using CNN and LSTM-Based Deep Learning Models. 2020.

[21] S. Kim and M. Kang, "Financial series prediction using Attention LSTM," Papers, Art. no. 1902.10877, Feb. 2019, Accessed: Nov. 10, 2023. [Online]. Available: https://ideas.repec.org//p/arx/papers/1902.10877.html

[22] A. Yadav, C. K. Jha, and A. Sharan, "Optimizing LSTM for time series prediction in Indian stock market," Procedia Comput. Sci., vol. 167, pp. 2091–2100, Jan. 2020, doi: 10.1016/j.procs.2020.03.257.

[23] A. Kia, "Using MLP and RBF Neural Networks to Improve thePrediction of Exchange Rate Time Series with ARIMA," Int. J. Inf. Electron. Eng., Jan. 2012, doi: 10.7763/IJIEE.2012.V2.157.

[24] H. Yan and H. Ouyang, "Financial Time Series Prediction Based on Deep Learning," Wirel. Pers. Commun., vol. 102, no. 2, pp. 683–700, Sep. 2018, doi: 10.1007/s11277-017-5086-2.

[25] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," Neural Comput. Appl., vol. 32, no. 23, pp. 17351–17360, Dec. 2020, doi: 10.1007/s00521-020-04867-x.

[26] D. Kobiela, D. Krefta, W. Król, and P. Weichbroth, "ARIMA vs LSTM on NASDAQ stock exchange data," Procedia Comput. Sci., vol. 207, pp. 3836–3845, Jan. 2022, doi: 10.1016/j.procs.2022.09.445.

[27] R. Xiao, Y. Feng, L. Yan, and Y. Ma, "Predict stock prices with ARIMA and LSTM." arXiv, Aug. 31, 2022. doi: 10.48550/arXiv.2209.02407.

[28] D. J. Pack, "In defense of ARIMA modeling," Int. J. Forecast., vol. 6, no. 2, pp. 211–218, Jul. 1990, doi: 10.1016/0169-2070(90)90006-W.

[29] [1] R. Fildes, M. Hibon, S. Makridakis, and N. Meade, "Generalising about univariate forecasting methods: further empirical evidence," Int. J. Forecast., vol. 14, no. 3, pp. 339–358, Sep. 1998, doi: 10.1016/S0169-2070(98)00009-0.

[30] L. J. Tashman, "Out-of-sample tests of forecasting accuracy: an analysis and review," Int. J. Forecast., vol. 16, no. 4, pp. 437–450, Oct. 2000, doi: 10.1016/S0169-2070(00)00065-0.

[31] R. Zhang et al., "Comparison of ARIMA and LSTM in Forecasting the Incidence of HFMD Combined and Uncombined with Exogenous Meteorological Variables in Ningbo, China," Int. J. Environ. Res. Public. Health, vol. 18, no. 11, Art. no. 11, Jan. 2021, doi: 10.3390/ijerph18116174.

[32] S. Siami-Namini and A. S. Namin, "Forecasting Economics and Financial Time Series: ARIMA vs. LSTM." arXiv, Mar. 16, 2018. doi: 10.48550/arXiv.1803.06386.

[33] T. Feng et al., "The comparative analysis of SARIMA, Facebook Prophet, and LSTM for road traffic injury prediction in Northeast China," Front. Public Health, vol. 10, 2022, Accessed: Nov. 10, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpubh.2022.946563.

[34] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," Eur. Phys. J. Spec. Top., vol. 228, no. 10, pp. 2313–2324, Oct. 2019, doi: 10.1140/epjst/e2019-900046-x.

[35] Z. Ding, R. Xia, J. Yu, X. Li, and J. Yang, "Densely Connected Bidirectional LSTM with Applications to Sentence Classification." arXiv, Feb. 02, 2018. doi: 10.48550/arXiv.1802.00889.

[36] S. Bharadwaj, R. Varun, P. S. Aditya, M. Nikhil, and G. C. Babu, "Resume Screening using NLP and LSTM," in 2022 International Conference on Inventive Computation Technologies (ICICT), Jul. 2022, pp. 238–241. doi: 10.1109/ICICT54344.2022.9850889.

[37] S. M. Bartolomei and A. L. Sweet, "A note on a comparison of exponential smoothing methods for forecasting seasonal series," Int. J. Forecast., vol. 5, no. 1, pp. 111–116, Jan. 1989, doi: 10.1016/0169-2070(89)90068-X.

[38] R J Hyndman - "Variations on rolling forecasts," Rob J Hyndman. July 16, 2014. Available: https://robjhyndman.com/hyndsight/rolling-forecasts/

[39] E. Dave, A. Leonardo, M. Jeanice, and N. Hanafiah, "Forecasting Indonesia Exports using a Hybrid Model ARIMA-LSTM," Procedia Comput. Sci., vol. 179, pp. 480–487, Jan. 2021, doi: 10.1016/j.procs.2021.01.031.