A comparison of the optimized LSTM , XGBOOST and ARIMA in Time Series forecasting

Iliana Paliari
Graduate of school of science
Hellenic Open University
Patras, Greece
paliari.iliana@gmail.com

Aikaterini Karanikola
Department of Mathematics
University of Patras
Patras, Greece
karanikola@upatras.gr

Sotiris Kotsiantis
Department of Mathematics
University of Patras
Patras, Greece
sotos@math.upatras.gr

***Abstract*— The term time series refers to historical data comprise of observations that are made in a fixed time step, successively, over a period of time. This work focuses on the training and application of modern Machine Learning approaches, like Deep Neural Network techniques, to model and predict general time series obtained from several open databases. The selection of the data that were used in the experiments was focused on specific economic and social phenomena, intending to predict their evolution over time. The main and final goal remains the comparison of the aforementioned predicting approaches, as well as the optimization of them in order to improve their accuracy.**

Keywords*— *Financial Forecasting; Long-Short-Term Memory (LSTM); XGBoost; ARIMA model; Australian Stock Market; Short-term Forecast; Stock Exchange.

I. Introduction

Making decisions and selecting the optimal alternative among the ones available, concerning the best outcome of a situation, is a task that individuals face daily. Given the abundance of available data, as well as the powerful tools that the fields of Statistics and Machine Learning provide, it is expected that the field of forecasting has recently attracted a vast amount of interest. In economics, making accurate predictions about several financial phenomena may be beneficial for various reasons. Predicting the closing value for the stock market is an example of such a financial forecasting task, that due to its complex nature is not only difficult to accomplish but also crucial, since building accurate and robust predictive models may have an important role in terms of shareholder purchases and sales. A vast amount of research is focused on the predictions of the stock market fluctuation, price change direction, portfolio asset allocation, and returns. Statistical techniques include, among others, ARCH (Autoregressive Conditional Heteroscedasticity) [1], ARIMA (Autoregressive Integrated Moving Average) [2], and GARCH (Generalized Autoregressive Conditional Heteroskedasticity) [3]. The widely known family of ARIMA models, noted for their robustness and efficiency, have been extensively used in such forecasting scenarios [4][5][6]. On the other hand, modern alternative approaches as Recurrent Neural Networks (RNNs) and Long short-term memory (LSTM) RNNs have also been used for the same purpose [7] with remarkable results [8][9][10].

The goal of this work is to exploit the use of ARIMA [11], LSTM [12], and XGBOOST [13] models in short-term stock price prediction on real-world datasets, consisting of stock market financial data. The performance of such methods may benefit investors by providing accurate predictions, which can lead to making optimal decisions and maximizing their profits.

This paper is structured as follows: in the following section, previous research is provided regarding time-series forecasting related to financial data. The description of prediction techniques that are used in the experimental procedure, along with the corresponding results, are presented respectively in sections III and IV. Finally, in the last section, the conclusions of our work are discussed and future extensions for further investigation are suggested.

II. Related work

Time series forecasting has a rich amount of related literature due to its popularity in recent years. Aspects of previous work related to the subject of this paper will be briefly discussed in this section, with an emphasis on recent research on time series forecasting on financial data, as well as works aiming to compare different forecasting methods and reach conclusions about their performance and limitations. The task of correctly predicting the evolution of the outcome of financial procedures has been an appealing field of interest among researchers, as the economy not only forms the relations between groups of people and even countries but also is the main index of the prosperity of contemporary society. The stock market's evolution has played a key role in defining modern financial environments, and the arrival of an increasing number of companies and organizations into the stock market has had a substantial impact on recent economic growth. Many different approaches, as well as specific models, have been proposed over the previous years, for predicting not only time series related to financial tasks, but also time series related to a wide range of different phenomena. Statistical methods, like the ARIMA models, dominated the field for the previous decades, and have been used extensively for predicting the closing value of stock markets [14]. ARIMA models also seem to perform fairly well, so they have a strong potential in short-term prediction [15][16]. Moreover, ARIMA has also been used in combination with other forecasting techniques, forming new hybrid methods. A hybrid ARIMA- Back Propagation Neural Network (BPNN) model for time series prediction of the Chinese stock market was presented in [17], while a novel approach that integrates ARIMA with the Empirical Wavelet Transform (EWT) technique, along with improved Artificial Bee Colony (ABC) algorithm and called the Extreme Learning Machine (ELM) Neural Network was proposed in [18]. ARIMA models have been also combined and formed accurate hybrid methods with GARCH [19], Support Vector Machines (SVMs) [20], and multiple Machine Learning methods like Multi-Layer Perceptron (MLP) and Support Vector Regression (SVR) [21].

Pure Machine Learning methods have also been utilized to tackle the problem of stock market forecasting [22][23]. The Principal Component Analysis and Recurrent Neural Network (RNNs) were integrated to forecast the stock price of Casablanca stock exchange, to reduce the number of features in the dataset [24], while RNNs have also been recently used in forecasting Sri Lankan Stock Market, demonstrating remarkable results [25]. LSTNs were employed to predict future trends of stock prices based on the price history, alongside technical analysis indicators in [8], while sentiment analysis has also been exploited in combination with LSTMs schemes, increasing significantly the predictive performance of the model [26]. Attention-based deep learning models [27], as well as bidirectional and stacked LSTM predictive models [28] have been exploited for forecasting the movement of stock markets.

Given that this research aims to compare various algorithms for forecasting time series and draw useful conclusions, it would be a huge oversight not to provide references to studies that make similar comparisons. In [29] a comparison between the performance of traditional ARIMA schemes and LSTMs is presented, while in [30] a comparison of different ARIMA models and Artificial Neural Networks (ANNs) is made, indicating that ANNs dominated the variations of ARIMA models in general. In [31] the performance of CNNs is compared to standard backpropagation NN with one hidden layer and LSTM networks, while in [32] a comparison of RNNs including an observational review using both LSTM and GRU networks is presented. Lastly, An insightful comparison between ANN, hybrid methods, and multiple regression techniques is presented in [33], whereas in [34] several machine learning methods are being compared.

III. A REVIEW OF THE EXAMINED MODELS

A. The ARIMA model

The acronym [35] ARIMA stands for Auto-Regressive Integrated Moving Average. ARIMA models aim to describe the autocorrelations in data. The letters of the acronym depict the philosophy behind this family of models. *AR* stands for Autoregressive, which implies that in this type of regression model, the dependent variable depends on its past values, or in other words that there is autocorrelation between current values and values in previous steps. *I* refers to Integrated, which means that data are stationary. A time series become stationary, by subtracting – or differencing - the observations from the previous values, as in this way the mean of the time series is stabilized, so seasonality and trends are eliminated or at least reduced. Concerning the above, stationary time series do not present predictable patterns in the long term, as their properties do not depend on the time at which the series is observed. Time series with trends and seasonality are not stationary. As autocorrelations remain constant over time, a random variable of this form can be viewed as a combination of signal and noise, and the signal could be a pattern of fast or slow mean reversion, or sinusoidal oscillation, or have a seasonal component. An ARIMA model can be considered as a "filter" that tries to separate the signal from the noise. After that, the signal is extrapolated to obtain future forecasts [21][29]. The last two letters *MA* stand for the dependency between an observation and a residual error from a moving average model applied to lagged observations. Thus, the output of the model depends on a linear way on previous observations, which also means that the forecasting errors are related linearly with past errors. This leads to one of the most important limitations of the model, which has to be taken into consideration. The reliability of the model depends strongly on historical data, so the number of available observations and their accuracy has a crucial role in the quality of the forecasts.

B. The Long Short – Term Memory Neural Network

Long Short-Term Memory networks [7] were firstly introduced by Hochreiter & Schmidhuber in 1997 [36]. In previous years, RRNs have been used in a variety of problems, including speech recognition [37], language modeling [38], and image captioning [39], but they tend to act poorly while data contain long-term dependencies, and suffer from the vanishing gradient problem. LSTMs are a specific kind of Recurrent Neural Networks (RNNs), designed to solve the vanishing gradient problem, by preserving the error that can be backpropagated through time and layers. Moreover, LSTMs overcome the weakness of learning long-range dependencies, a trait that is essential in time-series forecasting [40]. Unlike traditional neural networks, an LSTM's layer consists of a set of blocks, recurrently connected. Each block contains one or multiple recurrently connected memory cells and three multiplicative units – the input, output and, forget gates. The memory cell remembers values over arbitrary time intervals, while the three gates arrange the information flow into and out of the cell. The cell stores the desired information, by performing acts of reading, writing, and erasing, via the gates that open and close. These extra interactions among the four aforementioned elements of memory blocks, make LSTMs capable of learning which data are important to keep and which are not. Gates are associated with sets of weights, which filter the information blocking it, or allowing it to pass. Weights are not fixed but adjust over the recurrent learning process. Thus, relevant information can pass down the chain of sequences to make predictions, achieving to maintain and represent long-term data.

C. The XGBOOST algorithm

Gradient boosting is a powerful ensemble Machine Learning technique, that is used in classification and regression problems, while is also famous in predictive scenarios [41]. As an ensemble technique, gradient boosting combines the results of several weak learners, referred to as base learners, to build a model that performs generally better than the conventional single machine learning models. Typically, gradient boosting utilizes decision trees as base learners. Like other boosting methods, the core idea of gradient boosting is that during the learning procedure new models are build and fitted consecutively and not independently, to provide better predictions of the output variable. New base learners are constructed aiming to minimize a loss function, associated with the whole ensemble. Instances that are not predicted correctly in previous steps and score higher errors are correlated with larger weight values, so the model can focus on them, and learn from its mistakes.

XGBoost stands for Extreme Gradient Boosting and is a specific implementation of gradient boosting, that was initially developed at the University of Washington as a research project [42]. Specific details that differ in XGBoost than traditional gradient boosting techniques, make this implementation more powerful, resulting in better performance. Specifically, XGBoost uses advanced regularization which improves model generalization and reduces overfitting. Moreover, computes second-order gradients of the loss function, which provide more information about the gradient's direction, making it easiest to minimize the loss function. Other advantages of the implementation of XGBoost are that has an in-built capability to handle missing values, and also allows parallel processing, a fact that compensates the increased computational demand.

IV. Experimental procedure

A. *Description of the data*

One of the major challenges throughout the experimental procedure was the selection of the independent variables, as a proper selection can drastically impact the forecasting efficiency of the model. The data used in the experimental procedure were daily stock market prices of six Australian Corporations, forming six distinct datasets, which from now on will be referred to, with the following abbreviations: AX1, K2F, KMD, OKJ, ORE, and RHC. The input data cover the period from January 24 of 2014 to January 24 of 2020, having a total number of 1521 observations each. Six different attributes were recorded for each observation, which represents a day, which are noted by the names "Open", "High", "Low", "Close", "Adj Close", and "Volume traded". "Open" represents the opening price of the index at the start of the corresponded trading day, while "Close" indicates the closing price of the stock at the end of the same day. "Low" and "High" represent the minimum and the maximum price of the stock during the day respectively. As the closing price of the stock reflects all the activity of the index throughout the trading day, "Close" was chosen as the target value to be predicted.

B. *Development of the models*

1) ARIMA(p,d,q) Model Development: Concerning the form of the data that were used in the experimental procedure, it was observed that the original pattern of the time series of the index is not stationary. The time series have random walk pattern and vary randomly, and no global trend or seasonality pattern was observed. For implementing the ARIMA model the values of the parameters q, d and q had to be set. The parameters of the ARIMA model, which are related to the components AR, I, and MA indicate the type of the model and are defined as follows:

- p: The number of lag observations included in the model, also called the lag order.
- d: The number of times that the raw observations are differenced, also called the degree of differencing.
- q: The size of the moving average window, also called the order of moving average.

First, the parameters p, d, and q were set to 5, 1, and 0 respectively, leading to an ARIMA(5,1,0) model. As the original time series was not stationary is essential not to set d equal to zero, in order to transform it into a stationary one. The parameter d was set to 1 as the original series has a random walk pattern with an almost constant average trend. Finally, predictions were made by calling the predict() function and specifying the index of times to be predicted. Furthermore, the predict() function was used on the ARIMA results object to make predictions. To test the efficiency of the model, three error metrics were calculated: Root of Mean Squared Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE).

Classical regression models such as the one that was built and used in this section, are widely used in many forecasting tasks, like for electric power forecasting [43][44], which has been studied extensively in and It has been identified that classical regression is often insufficient for efficiently explaining the dynamics of a time series. An interesting approach was the introduction of correlations that may be generated through lagged linear relations. This led to the Autoregressive (AR) and Autoregressive Moving Average (ARMA) models. Adding nonstationary models to the mix led to the Autoregressive Integrated Moving Average (ARIMA) model, and that states for the landmark work by Box and Jenkins [45].

2) ANN-LSTM Model Development: In order to apply the LSTM neural network, the Keras library [46] in python was used. The initial datasets were normalized in [0,1] interval, by using the MinMax Scaler preprocessing class from the scikit-learn library [47]. A python function was created to split each dataset into two parts. The first one consisted of 80% of the observations, was used as a train set to train the model, while the second one, which consisted of the remaining 20% of the observations, was used as a test set. This function used two input arguments: dataset and look_back. The first one is a Numpy [48] array that represents the initial dataset, while the latter represents the number of previous time steps to be used as input variables to predict the following value in the unit of time. In our case the default value of look_back was equal to one, allowing us to investigate the case of given the first X observations at corresponding time t, the output Y of the model is the prediction of the value by the time t+1. Concerning the topology of our network, our model utilizes one visible layer with one input, a hidden layer with four LSTM neurons, and an output layer, which produces the single value prediction. The linear function was used as kernel function, while callbacks have also been inserted to speed up the procedure of determining the procedure of the best number of epochs. Adam was also used as an optimizer and the model was trained for 250 epochs, using feedforward algorithm.

3) XGBoost model development: To implement the gradient boosting decision tree XGBoost python library [49] was used. Several parameters must be initialized before the execution of the method, which define the model, and they are referred to as hyperparameters. Hyperparameters belong to three categories:

- General hyperparameters: relate to the functioning of the algorithm and define the kind of booster that is used, which is usually a tree of linear. In our case gbtree was used, to utilize tree-based models.
- Booster hyperparameters: relate to the chosen type of booster. For example, concerning gbtree several parameters define different information out of it, such as its depth, etc.
- Learning task hyperparameters: define the learning scenario. In our case, the optimization objective is the logistic regression, while the final output is the

predicted value y. The loss function which controls the predictive power and needs to be optimized is the root mean squared error. The way new nodes are added to the tree is selected as grow_policy=lossguide , which splits at nodes with higher loss change.

Note that before performing the gradient boosting procedure, the values of the datasets were once again normalized, by using the MinMax Scaler preprocessing class from the scikit-learn library [47].

V. RESULTS AND DISCUSION

To test the performance of the examined methods, three different evaluation metrics were used. Taking into account that there is not a single performance metric that can guarantee safe results, as each of them has both advantages and disadvantages, we used three well-known and widely used for this purpose in literature methods. These are Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). With the assumption that a is the actual value of an observation, and p is the predicted one, the formulas for calculating the three metrics are being presented in Table I.

TABLE I
EVALUATION METRICS

Evaluation Metric	**Formula**
MSE	$\frac{1}{n}\sum_{i=1}^{n}(a_i - p_i)^2$
RMSE	$\sqrt{\frac{\sum_{i=1}^{n}(a_i - p_i)^2}{n}}$
MAE	$\sum_{i=1}^{n}\lvert a_i - p_i\rvert^2$

Considering the results of the three evaluation metrics, which are presented in Table II, some interesting assumptions can be made. Note that ARIMA(5,1,0) model was designed as a benchmark for comparison. It is clear, that LSTM and XGBoost outperform ARIMA(5,1,0) in most of the cases, while ARIMA(5,1,0) achieves lower error scores only in two datasets: "K2F" and "OKJ". In these two stocks, the values are significantly lower, when compared to the others, a fact that seems to have an impact on the results of the experimental procedure, thus ARIMA(5,1,0) outperformed both ML methods in terms of incorporating the magnitude of features without neglecting the units. On the other hand, LSTM seems to outperform the other two methods, achieving the lowest scores, in most of the examined datasets. The LSTM model scores its best predictions in "ORE" stock, while scores its worst in "K2F". XGboost model achieves its best performance on "RCH" and its worst in "K2F", in which ARIMA beats both its rivals, like in "OKJ" dataset. In general, the performance of the LSTM model is proven fairly good and slightly better than the performance of the XGBoost. The results of the experimental procedures are also depicted in Figures 1, 2, and 3, concerning the three forecasting methods, providing useful insight into results. The true and the predicted values are depicted in the same frames for each dataset, presenting the overall performance of the methods. In all cases the predicted values are close to the real ones, indicating the good performance of the models. At this point, it would also be mentioned that the Exploratory Data Analysis of each stock showed a positive linear trend in each dataset, which is a favorable trait while forecasting upcoming prices.

Before moving to the conclusions, we should also mention some interesting remarks that emerged during the experimental procedure, as well as the visualization of the results. First, during the period from the 8th of September of 2016 to the 20th of June of2019 there was a fixed price of the closing value of ‘OKJ’. After a thorough consideration of the “Annual Financial” report of the “‘The Oakajee Corporation Limited’” company, we identified that no shareholders were attracted by the OKJ stock within the years 2016, 2017, 2018. New investors of OKJ were listed during 2019 (apart from the 20 largest shareholders of ordinary shares). According to the dataset, this is when the price of the stock changes for the first time since 08/09/2016. Another interesting fact that may have an impact on the results, is that ‘The Oakajee Corporation Limited’, and the “Paynes Find Gold Project” cover 112 km^2 of ground within the Payne’s Find Greenstone Belt in Western Australia, which was affected by the massive fire this period. As a result, the price of the stock faces a dramatic decrease. Both the aforementioned facts imply how various unexpected factors can result in dramatic changes in stock market prices, affecting the performance of predictive models.

TABLE II
RESULTS OF THE EXPERIMENTAL PROCEDURE

Dataset	**Algorithm**	**Metrics**		
		RMSE	***MSE***	***MAE***
AX1	XGBoost	0.02943	0.00087	0.02275
	LSTM	**0.02573**	**0.00066**	**0.01906**
	ARIMA	0.03642	0.00132	0.02644
K2F	XGBoost	0.03997	0.00160	0.02768
	LSTM	0.02728	0.00074	0.01728
	ARIMA	**0.01183**	**0.00013**	**0.00805**
KMD	XGBoost	0.03550	0.00126	0.02359
	LSTM	**0.02163**	**0.00047**	**0.01419**
	ARIMA	0.05618	0.00315	0.03734
OKJ	XGBoost	0.06316	0.00399	0.05798
	LSTM	0.02438	0.00059	0.00825
	ARIMA	**0.00145**	**0.000002**	**0.00039**
ORE	XGBoost	0.01982	0.00039	0.01528
	LSTM	**0.01833**	**0.00034**	**0.01370**
	ARIMA	0.14266	0.02035	0.10311
RHC	XGBoost	0.02211	0.00049	0.01687
	LSTM	**0.02050**	**0.00042**	**0.01494**
	ARIMA	0.88528	0.78372	0.63510

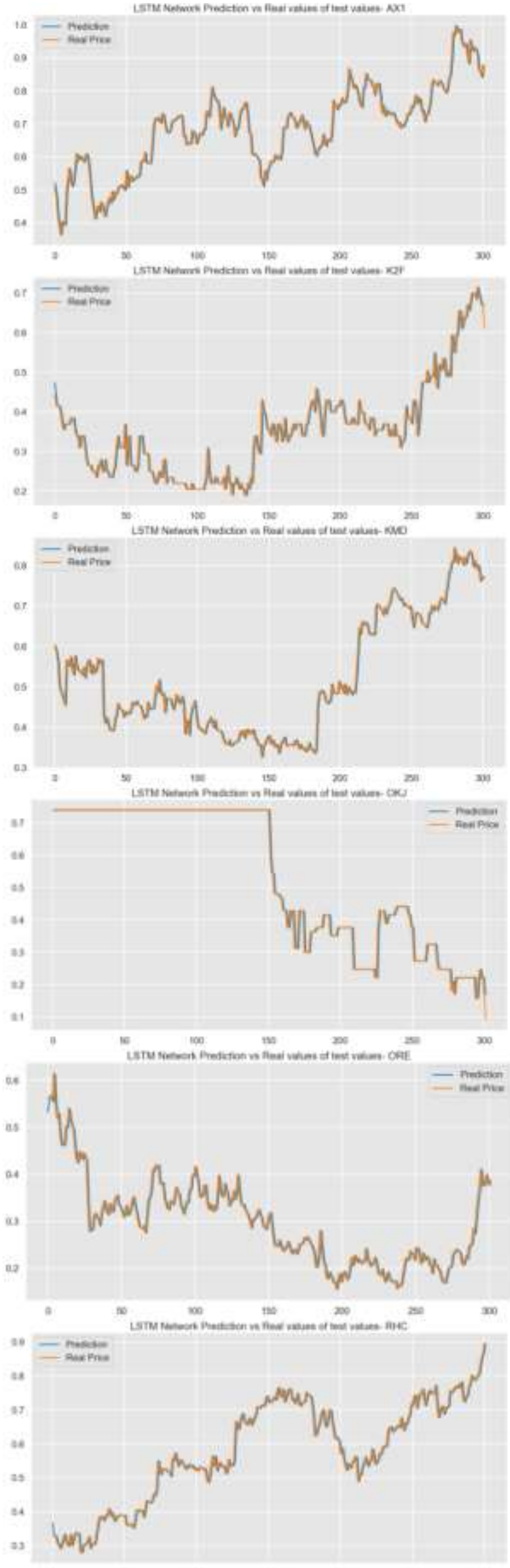

Figure 1: Predicted vs true values using LSTMs

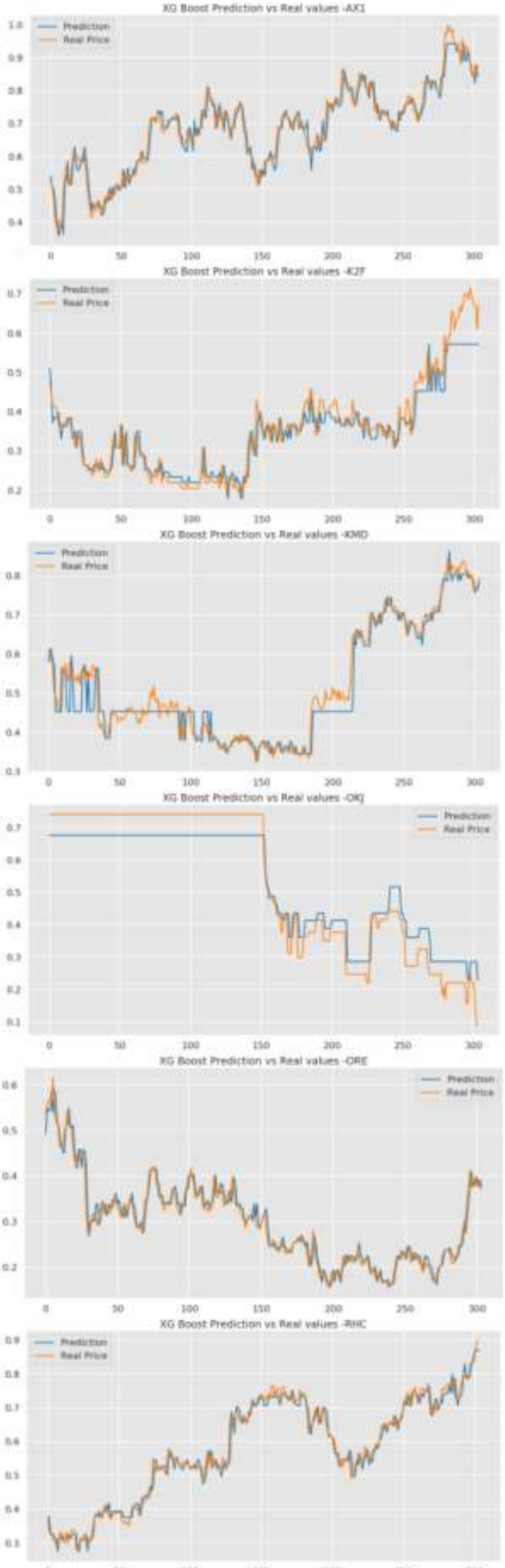

Figure 2:Predicted vs true values using XGBoost

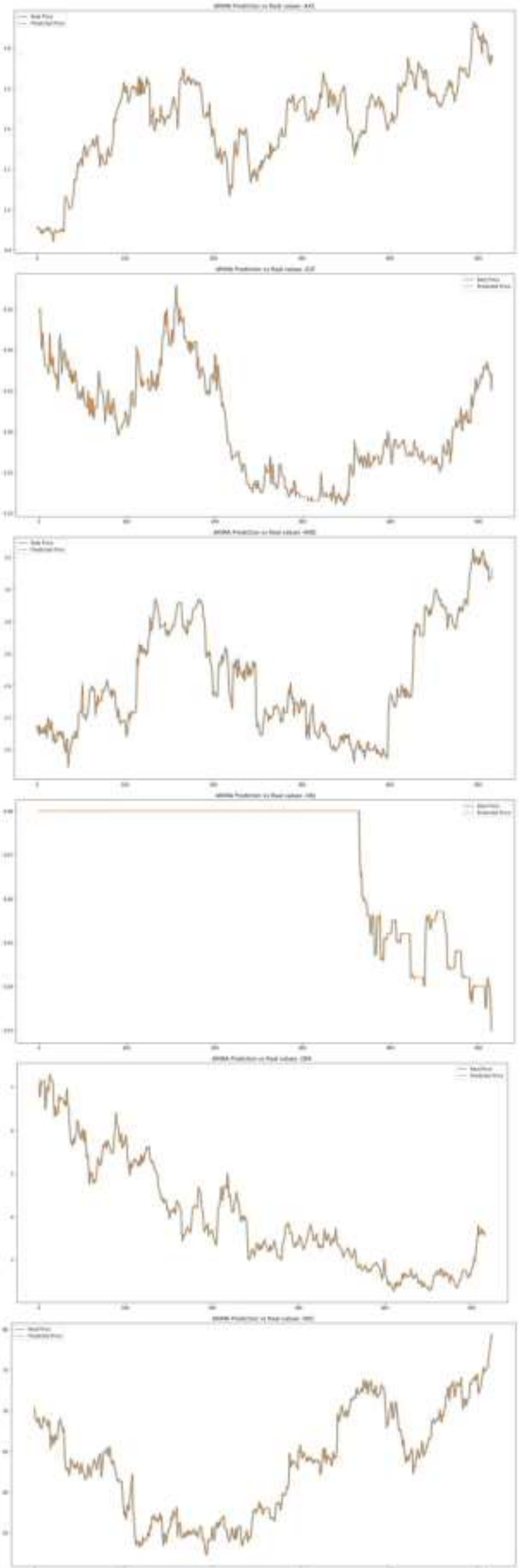

Figure 3:Predicted vs true values using ARIMA

VI. CONCLUSIONS

In this work, the feasibility of applying modern ML approaches, such as LSTM and XGBoost on forecasting financial time series data was examined. For this purpose, datasets comprised of Australian stock market data were utilized. Considering the advantages and disadvantages of both methods, we reached conclusions about the selection of the best approach. Furthermore, optimization for both methods took place. The outcome of the experimental procedure leads to the result that both modern ML models, outperform the benchmark ARIMA model, concerning all comparative metrics, except the case that the closing values are extremely low. The results of our work suggest that modern ML techniques not only can be used safely in stock market prediction, but they also tend to outperform traditional statistical approaches that have dominated the field for decades.

The process of decision-making in financial domain fields can be significantly affected the criteria that define the direction of prediction [50]. The extensive incorporation of the Exploratory Data Analysis aims to make better short-term and long-term decisions. Their combination, along with the definition of better direction-prediction criteria may lead to favorable results in financial research. Concerning future extensions of this work, comparisons of different markets and stock exchanges will be examined. Furthermore, the benefits of ensemble learning theory [51][52] would be exploited in prediction tasks, by developing a hybrid model that utilizes ensembles in order to combine the weighted outputs of LSTM and XGBoost, which would increase the prediction accuracy of the aforementioned models. An extended comparison of the hybrid model with ARIMA and traditional LSTM and XGBoost, will also take place to test its performance.

REFERENCES

[1] R. F. Engle, "Autoregressive Conditional Heteroscedacity with Estimates of variance of United Kingdom Inflation,journal of Econometrica, Volume 50, Issue 4 (Jul., 1982),987-1008.," *Econometrica*, vol. 50, no. 4. pp. 987–1008, 1982.

[2] S. AL Wadi, M. Almasarweh, and A. A. Alsaraireh, "Predicting Closed Price Time Series Data Using ARIMA Model," *Mod. Appl. Sci.*, vol. 12, no. 11, p. 181, 2018, doi: 10.5539/mas.v12n11p181.

[3] R. T. Baillie, T. Bollerslev, and H. O. Mikkelsen, "Fractionally integrated generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 74, no. 1. pp. 3–30, 1996, doi: 10.1016/S0304-4076(95)01749-6.

[4] R. G. Kavasseri and K. Seetharaman, "Day-ahead wind speed forecasting using f-ARIMA models," *Renew. Energy*, vol. 34, no. 5, pp. 1388–1393, 2009, doi: 10.1016/j.renene.2008.09.006.

[5] D. Eni and F. J. Adeyeye, "Seasonal ARIMA Modeling and Forecasting of Rainfall in Warri Town, Nigeria," *J. Geosci. Environ. Prot.*, vol. 03, no. 06, pp. 91–98, 2015, doi: 10.4236/gep.2015.36015.

[6] G. Bandyopadhyay, "Gold Price Forecasting Using ARIMA Model," *J. Adv. Manag. Sci.*, no. July 2018, pp. 117–121, 2016, doi: 10.12720/joams.4.2.117-121.

[7] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, 2019, doi: 10.1016/j.neucom.2018.09.082.

[8] D. M. Q. Nelson, A. C. M. Pereira, and R. A. De Oliveira, "Stock Market's Price Movement Prediction WithLSTM Neural Networks," *IEEE Trans. Neural Networks*, vol. 21, no. 8, pp. 1378–1378, 2010, doi: 10.1109/tnn.2010.2063350.

[9] A. M. Rather, A. Agarwal, and V. N. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3234–3241, 2015, doi: 10.1016/j.eswa.2014.12.003.

[10] Q. Zhuge, L. Xu, and G. Zhang, "LSTM neural network with emotional analysis for prediction of stock price," *Eng. Lett.*, vol. 25, no. 2, pp. 167–175, 2017.

[11] Y. Lai and D. A. Dzombak, “Use of the autoregressive integrated moving average (Arima) model to forecast near-term regional temperature and precipitation,” *Weather Forecast.*, vol. 35, no. 3, pp. 959–976, 2020, doi: 10.1175/WAF-D-19-0158.1.

[12] A. Essien and C. Giannetti, “A Deep Learning Framework for Univariate Time Series Prediction Using Convolutional LSTM Stacked Autoencoders,” *IEEE Int. Symp. Innov. Intell. Syst. Appl. INISTA 2019 - Proc.*, pp. 1–6, 2019, doi: 10.1109/INISTA.2019.8778417.

[13] A. Kadiyala and A. Kumar, “Applications of python to evaluate the performance of decision tree-based boosting algorithms,” *Env.. Prog. Sus.. Energy*, vol.37, no.2, pp.618–623, 2018, doi: 10.1002/ep.12888.

[14] Y. Dong, S. Li, and X. Gong, “Time Series Analysis: An application of ARIMA model in stock price forecasting,” vol. 29, no. Iemss, pp. 703–710, 2017, doi: 10.2991/iemss-17.2017.140.

[15] A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo, “Stock price prediction using the ARIMA model,” *Proc. - UKSim-AMSS 16th Int. Conf. Comput. Model. Simulation, UKSim 2014*, pp. 106–112, 2014, doi: 10.1109/UKSim.2014.67.

[16] M. Almasarweh and S. AL Wadi, “ARIMA Model in Predicting Banking Stock Market Data,” *Mod. Appl. Sci.*, vol. 12, no. 11, p. 309, 2018, doi: 10.5539/mas.v12n11p309.

[17] L. Xiong and Y. Lu, “Hybrid ARIMA-BPNN model for time series prediction of the Chinese stock market,” *2017 3rd Int. Conf. Inf. Manag. ICIM 2017*, pp. 93–97, 2017, doi: 10.1109/INFOMAN.2017.7950353.

[18] H. Yu, L. J. Ming, R. Sumei, and Z. Shuping, “A Hybrid Model for Financial Time Series Forecasting-Integration of EWT, ARIMA with the Improved ABC Optimized ELM,” *IEEE Access*, vol. 8, pp. 84501–84518, 2020, doi: 10.1109/ACCESS.2020.2987547.

[19] F. H. Mustapa and M. T. Ismail, “Modelling and forecasting S&P 500 stock prices using hybrid Arima-Garch Model,” *J. Phys. Conf. Ser.*, vol. 1366, no. 1, 2019, doi: 10.1088/1742-6596/1366/1/012130.

[20] 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, 2005, doi: 10.1016/j.omega.2004.07.024.

[21] D. S. Domingos, J. F. L. de Oliveira, and P. S. G. de Mattos Neto, “An intelligent hybridization of ARIMA with machine learning models for time series forecasting,” *Knowledge-Based Syst.*, vol. 175, pp. 72–86, 2019, doi: 10.1016/j.knosys.2019.03.011.

[22] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, “Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; A Comparative Analysis,” *IEEE Access*, vol. 8, pp. 150199–150212, 2020, doi: 10.1109/ACCESS.2020.3015966.

[23] A. Sharma, D. Bhuriya, and U. Singh, “Survey of stock market prediction using machine learning approach,” *Proc. Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2017*, vol. 2017-January, pp. 506–509, 2017, doi: 10.1109/ICECA.2017.8212715.

[24] Z. Berradi and M. Lazaar, “Integration of Principal Component Analysis and Recurrent Neural Network to Forecast the Stock Price of Casablanca Stock Exchange,” *Procedia Comput. Sci.*, vol. 148, pp. 55–61, 2019, doi: 10.1016/j.procs.2019.01.008.

[25] A. J. P. Samarawickrama, T. G. I. Fernando, S. Lanka, A. J. P. Samarawickrama, and T. G. I. Fernando, “A Recurrent Neural Network Approach in Predicting Daily Stock Prices,” *2017 IEEE Int. Conf. Ind. Inf. Syst.*, pp. 1–6, 2017, Available online: http://ieeexplore.ieee.org/document/8300345/.

[26] Z. Jin, Y. Yang, and Y. Liu, “Stock closing price prediction based on sentiment analysis and LSTM,” *Neural Comput. Appl.*, vol. 32, no. 13, pp. 9713–9729, 2020, doi: 10.1007/s00521-019-04504-2.

[27] L. C. Cheng, Y. H. Huang, and M. E. Wu, “Applied attention-based LSTM neural networks in stock prediction,” *Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018*, pp. 4716–4718, 2019, doi: 10.1109/BigData.2018.8622541.

[28] K. A. Althelaya, E. S. M. El-Alfy, and S. Mohammed, “Evaluation of Bidirectional LSTM for Short and Long-Term Stock Market Prediction,” *2018 9th Int. Conf. Inf. Commun. Syst. ICICS 2018*, vol. 2018-January, pp. 151–156, 2018, doi: 10.1109/IACS.2018.8355458.

[29] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, “A Comparison of ARIMA and LSTM in Forecasting Time Series,” *Proc. - 17th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2018*, pp. 1394–1401, 2019, doi: 10.1109/ICMLA.2018.00227.

[30] V. S. Pandey and A. Bajpai, “Predictive Efficiency of ARIMA and ANN Models: A Case Analysis of Nifty Fifty in Indian Stock Market,” *Int. J. Appl. Eng. Res.*, vol. 14, no. 2, pp. 232–244, 2019.

[31] I. Koprinska, D. Wu, and Z. Wang, “Convolutional Neural Networks for Energy Time Series Forecasting,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 2018-July, 2018, doi: 10.1109/IJCNN.2018.8489399.

[32] G. P. Zhang, “Neural networks for time-series forecasting,” *Handb. Nat. Comput.*, vol. 1–4, pp. 461–477, 2012, doi: 10.1007/978-3-540-92910-9_14.

[33] C. A. Ярушев, S. A. Yarushev, A. N. Averkin, and A. A.N., “Review of studies on time series forecasting based on hybrid methods, neural networks and multiple regression” *International J. "Programmnye produkty i Sist.*, vol. 31, no. 113, pp. 75–82, 2016, doi: 10.15827/0236-235x.113.075-082.

[34] G. Papacharalampous, H. Tyralis, and D. Koutsoyiannis, “Univariate Time Series Forecasting of Temperature and Precipitation with a Focus on Machine Learning Algorithms: a Multiple-Case Study from Greece,” *Water Resour. Manag.*, vol. 32, no. 15, pp. 5207–5239, 2018, doi: 10.1007/s11269-018-2155-6.

[35] M. Geurts, G. E. P. Box, and G. M. Jenkins, “Time Series Analysis: Forecasting and Control,” *J. Mark. Res.*, vol. 14, no. 2, p. 269, 1977, doi: 10.2307/3150485.

[36] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[37] A. Graves, N. Jaitly, and A. Mohamed, “HYBRID SPEECH RECOGNITION WITH DEEP BIDIRECTIONAL LSTM Alex Graves , Navdeep Jaitly and Abdel-rahman Mohamed Univ. of Toronto Dept. of Com. Science 6 King ’ s College Rd . Toronto , M5S 3G4 , Canada,” pp. 273–278, 2013.

[38] G. Kurata, B. Ramabhadran, G. Saon, and A. Sethy, “Language modeling with highway LSTM,” *arXiv*, pp. 244–251, 2017.

[39] C. Wang, H. Yang, and C. Meinel, “Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning,” *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 14, no. 2s, pp. 988–997, 2018, doi: 10.1145/3115432.

[40] T. Ese N *et al.*, “Long Short-Term Memory in Recurrent Neural Networks,” no. June, 2366, doi: 10.5075/epfl-thesis-2366.

[41] E. A. Madrid and N. Antonio, “Short-term electricity load forecasting with machine learning,” *Inf.*, vol. 12, no. 2, pp. 1–21, 2021, doi: 10.3390/info12020050.

[42] H. Zheng, J. Yuan, and L. Chen, “Short-Term Load Forecasting Using EMD-LSTM neural networks with a xgboost algorithm for feature importance evaluation,” *Energies*, vol. 10, no. 8, 2017, doi: 10.3390/en10081168.

[43] E. Yukseltan, A. Yucekaya, and A. H. Bilge, “Forecasting electricity demand for Turkey: Modeling periodic variations and demand segregation,” *Appl. Energy*, vol. 193, pp. 287–296, 2017, doi: 10.1016/j.apenergy.2017.02.054.

[44] D. H. Vu, K. M. Muttaqi, and A. P. Agalgaonkar, “A variance inflation factor and backward elimination based robust regression model for forecasting monthly electricity demand using climatic variables,” *Appl. Energy*, vol. 140, pp. 385–394, 2015, doi: 10.1016/j.apenergy.2014.12.011.

[45] G. C. T. Box, G.E.; Jenkins, G.M.; Reinsel, *Time Series Analysis: Forecasting and Control*, Vol. 734. NJ, USA: John Wiley & Sons: Hoboken, 2011.

[46] F. Chollet and others, “Keras.” GitHub, 2015, [Online]. Available: https://github.com/fchollet/keras.

[47] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in {P}ython,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[48] C. R. Harris *et al.*, “Array programming with {NumPy}” *Nature*, vol. 585,no.7825, pp.357–362, Sep.2020, doi:10.1038/s41586-020-2649-2.

[49] T. Chen and C. Guestrin, “{XGBoost}: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[50] M. Ballings, D. Van Den Poel, N. Hespeels, and R. Gryp, “Evaluating multiple classifiers for stock price direction prediction,” *Ex. Syst. Appl.*, vol. 42, no. 20, pp. 7046–7056, 2015, doi: 10.1016/j.eswa.2015.05.013.

[51] T. G. Dietterich, “Ensemble Methods in Machine Learning,” *Mult. Classif. Syst.*, vol. 1857, pp. 1–15, 2000, doi: 10.1007/3-540-45014-9.

[52] O. Sagi and L. Rokach, “Ensemble learning: A survey,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, pp. 1–18, 2018, doi: 10.1002/widm.1249.