IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 January 2024, accepted 15 March 2024, date of publication 25 March 2024, date of current version 29 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3381516

RESEARCH ARTICLE

Conditional Forecasting of Bitcoin Prices Using Exogenous Variables

ADEL MAHFOOZ AND JOSHUA L. PHILLIPS
Department of Computer Science, Middle Tennessee State University, Murfreesboro, TN 37132, USA
Corresponding author: Adel Mahfooz (am2fq@mtmail.mtsu.edu)

ABSTRACT Bitcoin is known for its high volatility, which makes it challenging to accurately predict future prices. In this study, we aim to forecast Bitcoin prices for a month by incorporating exogenous variables, specifically the interest rate and recession probability. Our primary objective is to explore whether these variables have a positive impact on the prediction of Bitcoin prices. We used two popular time series forecasting models: Long Short-Term Memory (LSTM) and Facebook Prophet. Our approach involves exploring the impact of these exogenous variables on the performance of the models and comparing their results through plots and cross-validation. We trained the models using historical Bitcoin price data along with exogenous variables and evaluated their performance on a test dataset. Our results indicate that LSTM outperforms Facebook Prophet in terms of Bitcoin price prediction accuracy. This is because, while Facebook Prophet is optimized for statistical forecasting modeling, LSTM has the capability to learn intricate patterns and relationships given the right architecture with sufficient neurons. Importantly, we demonstrate that incorporating interest rates and recession probabilities significantly enhances the predictive capability of our models. Our findings suggest that changes in interest rates and recession probabilities have an impact on Bitcoin prices, and our models perform better when equipped with this valuable information.

INDEX TERMS Bitcoin, cryptocurrency, exogenous variables, forecasting, interest rate, LSTM, machine learning, recession probability, time series.

I. INTRODUCTION

The cryptocurrency market has seen a surge in popularity and interest, with Bitcoin being the most well-known and widely used. As with any financial asset, accurate forecasting of Bitcoin prices is of great importance for investors and traders. This importance is underscored by the role traditional financial indicators play in influencing investor's decisions in a market known for its high volatility. Specifically, interest rates, set by central banks, directly affect borrowing decisions and, consequently, investments in high-risk assets like Bitcoin. Lower interest rates generally encourage such investments, while higher rates can reduce Bitcoin's appeal. Similarly, recession probability, an indicator of economic downturns, can prompt investors to shift away from riskier assets like Bitcoin in favor of more stable investments, thereby affecting its market value.

In this study, we explore the use of two different models, Long Short-Term Memory (LSTM) [1] and Facebook Prophet [2], to predict Bitcoin prices, incorporating these critical financial indicators as exogenous variables. Our aim is to analyze their impact on Bitcoin price forecasting, and to compare the performances of the LSTM and Prophet models. The interest rate is the overnight exchange rate between depository institutions for federal funds (balances kept at Federal Reserve Banks) [3]. On the other hand, the index of industrial production, real personal income excluding transfer payments, real manufacturing and trade sales, and non-farm payroll employment are four coincident monthly variables used to calculate recession probabilities in the United States [4]. We examine the effect of incorporating these variables on the accuracy of the models and, how each model handles the absence of changepoints in the

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Huei Cheng.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

exogenous variables. Our analysis demonstrates that although both models can make fair predictions in the presence of changepoints in exogenous variables, the LSTM model outperforms the Prophet model when there are no changepoints. This is because of its ability to learn complex patterns and relationships, making it more adaptable to changing market conditions.

Our study highlights the importance of considering exogenous variables in Bitcoin price forecasting and provides insights into the strengths and weaknesses of different modeling approaches. We hope that our findings will contribute to the development of more accurate and reliable models for predicting Bitcoin prices, and ultimately aid investors and traders in making informed decisions.

II. BACKGROUND

A. RELATED WORK

Time series forecasting is an important area of study in many fields, including finance and economics. The ability to accurately forecast future values of a time series can provide valuable insights and facilitate better decision-making. In recent years, the use of LSTM models has gained popularity in the field of time series forecasting because of their ability to handle long-term dependencies and their ability to model non-linear relationships.

Several studies have been conducted to predict established financial markets using neural networks [5], [6], [7], [8]. Traditional forecasting techniques, such as the Holt-Winters method, require the data to be separated into its components, such as trend, seasonal, and residual [9]. However, this method is most likely not appropriate for predicting Bitcoin price data since it is highly unlikely that a seasonal component is present in Bitcoin price fluctuations.

Prior studies on Bitcoin forecasting have been explored using different models, including comparisons. For instance, a head-to-head comparison between the auto-regressive integrated moving average (ARIMA) model and LSTM for Bitcoin was performed, where LSTM outperformed ARIMA by 1.40% mean absolute percent error (MAPE) [10]. In addition to Bitcoin, predictions for the day-ahead building level were made using deep learning models such as recurrent neural network (RNN) and, gated convolution neural network (CNN), and were compared with the seasonal auto regressive integrated moving average with the explanatory variable (ARIMAX) model, demonstrating that the 24-h gated CNN performed the best [11].

However, Bitcoin price fluctuations depend on many outside factors, such as market indices, gross domestic product (GDP), interest rates, sentiment, and recession. Several studies were performed that included these extra regressor variables in the models. For example, some researchers have used other currencies as extra regressors [12], [13]. To find the best suited one, they used a correlation matrix and found cryptocurrencies such as GBP, EUR, and JPY that best described the fluctuations in Bitcoin data [12]. Later, they incorporated these currencies in training and compared them with the ARIMA and prophet models, yielding accuracies 68% and 94.5% respectively. Other techniques included concatenating multiple parallel LSTM layers with N cryptocurrency data to predict one output [14]. Models such as the Bayesian optimized RNN and LSTM networks have also been used for forecasting [15]. The results were compared to those of the ARIMA model. Rather than focusing on one specific exchange, they took the average price from five major Bitcoin exchanges: Bitstamp, Bitfinex, Coinbase, OkCoin, and itBit. Other techniques, such as daily sentiment analysis using support vector machines (SVM), were also performed to improve the accuracy [14], [16]. To do so, they included the number of Bitcoin searches from Wikipedia (wikiviews) and Google (googleviews) and included public comments from Twitter posts (ntweets) as external variables.

These studies convey the message that, when predicting the highly volatile nature of certain financial data (such as Bitcoin), considering outside factors is necessary. To the best of our knowledge, outside factors such as the interest rate and recession probability have not yet been used in predicting Bitcoin prices. We hypothesize that incorporating these two factors might be helpful as the recession of year 2020 likely had an impact on the rise in Bitcoin price. Again, the current downfall of Bitcoin occurs with a continuous rise in the interest rate. In this study, along with comparisons, we include recession probability and interest rate as extra regressor variables and analyze their impact on performance.

B. MODELS

1) LSTM

In our study, we use a tool known as LSTM, which is a deep learning technique, to forecast the Bitcoin price [1]. LSTMs are designed to handle long-term dependencies in time series data, which means they can effectively capture patterns in the data that span a large number of time steps, which makes them well suited for modeling the price of Bitcoin. It is designed to overcome the vanishing gradient problem, which makes them more stable and less prone to overfitting than traditional simple RNNs. The model is built with five components: the input gate, forget gate, cell state, output gate, and hidden state.

a: INPUT GATE

The input gate controls the amount of data that enters the memory cell. This gate utilizes a sigmoid function over a concatenation of the previous hidden state and current input. As the value of the sigmoid function varies from zero to one, zero allows nothing to go through, while one passes everything to the cell state.

b: FORGET GATE

The forget gate controls the information leaving the memory cell. It determines how much information is removed from memory. This gate comes into action after the input gate. It is used to prevent memory cells from retaining irrelevant

or outdated information, which may negatively impact the predictions made by the LSTM. In a time series forecasting task, for example, the forget gate is used to forget information about past observations that are no longer relevant to the current time step. This allows the LSTM to focus on the most relevant information and make more accurate predictions about future observations.

c: OUTPUT GATE

The output gate controls the information that leaves the memory cell and flows to downstream layers. It chooses how much data should be read from memory.

d: CELL STATE

Long-term information is stored in the LSTM internal memory or cell state. It revolves through the entire LSTM channel with minor linear interactions. In a time series forecasting task, for example, the cell state is updated at each time step as the network processes the historical data. It retains information about past observations, which can be used to make predictions regarding future observations. The cell state can be thought of as a ''running memory'' that is updated over time, allowing the LSTM to incorporate information from multiple time steps into a single representation that summarizes the information from the entire input sequence.

e: HIDDEN STATE

The hidden state in an LSTM network is a summary of the information stored in the memory cells over time. It is a vector representation of the internal memory of the LSTM at each time step, and it is used to capture the contextual information from the input sequence. In general, the hidden state of an LSTM can be thought of as a ''memory'' that can be updated, written to, and read from over time. It is a key component of the LSTM architecture that allows the network to store information and use it to make predictions.

2) FACEBOOK PROPHET

We also explore another model called Prophet, which was developed by Facebook [2]. It is based on a Bayesian structural time series model that allows for modeling trends, seasonality, and outside factors in the data. The Bayesian framework allows regularization and prior information to be incorporated into the model. This helps improve the robustness and accuracy of the models.

Bitcoin prices can be influenced by external events, such as interest rates, regulation changes, sentiments, and market fluctuations. Prophet includes a mechanism for handling outliers and changepoints, allowing it to adapt to unpredictable events and improve the accuracy of predictions.

C. PERFORMANCE METRIC

For the performance metric, we use root-mean-squared error (RMSE) and MAPE for all models. A known disadvantage of MAPE is that it creates undefined values when the actual value is zero. However, it is safe to utilize for our analysis because we do not have any such zero value for Bitcoin prices. Additionally, utilizing MAPE allows us to compare our findings with those of earlier studies that employed MAPE as a performance metric.

D. MODEL EVALUATION

Model evaluation is an important step in any machine learning project because it allows us to assess the accuracy and performance of the developed models. In this study, we evaluate our models using a non-teacher forcing approach, which has several advantages over the teacher-forcing technique.

Teacher-forcing is a training method where the model is provided with the correct output sequence at each time step during training. However, this technique suffers from the problem of exposure bias, where the model is trained on perfect input-output pairs and does not learn how to handle the errors that arise during prediction. This can result in poor generalization and misleading prediction results.

In contrast, non-teacher forcing technique, such as using the predicted output sequence from the previous time step as input to the next time step, enables the model to learn from its own errors and handle uncertainty better. This leads to better generalization and more correct prediction accuracy.

E. KERAS TUNER

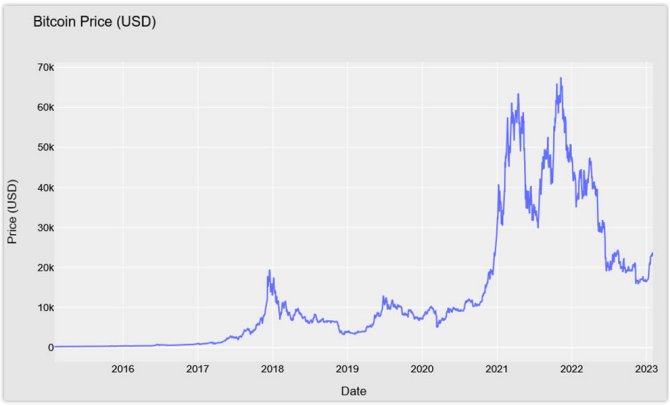
KerasTuner is a user-friendly and adaptable framework for optimizing hyperparameters, which simplifies the tedious and time-consuming task of hyperparameter search [17]. It enables users to define their search space with ease and choose from various built-in search algorithms such as Bayesian Optimization, Hyper-band, and Random Search to find the optimal hyperparameter values for models. There are two types of hyperparameter tuning: model hyperparameters and algorithm hyperparameters. Model hyperparameter tuning usually involves determining the number of hidden layers, number of neurons, activation functions etc. However, algorithm hyperparameters involve speeding up and improving the quality of learning by impacting the optimization process/algorithms.

F. KERAS EARLYSTOPPING

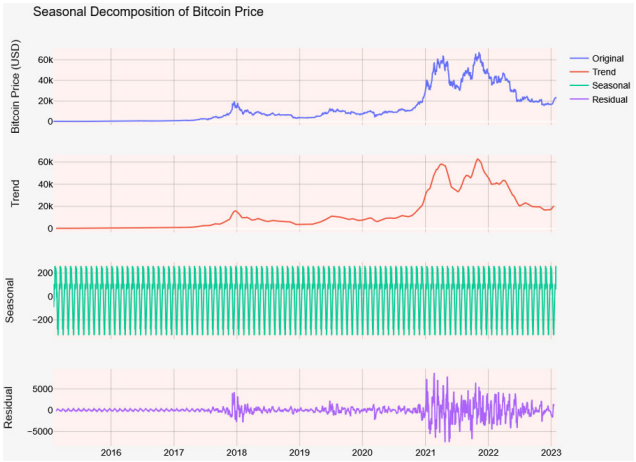
EarlyStopping is a callback function from the keras that can be utilized in the fitting process [18]. This callback function helps stop training depending on the *objective* and *mode* arguments passed to the callback. Mode usually takes 'min' or 'max' as a value, whereas the objective takes the name of the metric to observe. If a loss metric is used and a mode as 'min' is passed, it stops training if the metric is not seen to decrease any further. However, instead of immediate stopping, the function waits for a specific number of epochs. This parameter can be varied using the *patience* argument.

III. METHODS

We aim to ensure our work is reproducible by providing detailed information about our data collection and



(a) Historical closing price of Bitcoin in USD from February 2015 to January 2023, highlighting the cryptocurrency's high volatility. Noticeable are the large fluctuations in value, as indicated by the scale on the y-axis. The plot reveals no discernible pattern, underlining the unpredictability of Bitcoin's price movements over the observed period.



(b) Seasonal decomposition of Bitcoin price from February 2015 to January 2023, showing the original data, extracted trend, seasonal, and residual components. The trend line indicates alternating periods of rise and decline, accounting for a significant part of the movement. Seasonal fluctuations are consistent but confined within a Â±200 USD range, which is minor compared to the overall price scale that reaches up to 60K USD. The residuals exhibit no clear pattern, suggesting that price changes are likely driven by external, non-seasonal factors.

FIGURE 1. Comprehensive analysis of bitcoin and time series decomposition.

preprocessing methods as well as our modeling and analysis techniques in the following subsections.

A. TIME SERIES ANALYSIS

Time series analysis involves breaking down a dataset into its individual components, which include trends, seasonality, and residual components. Understanding these components is important before modeling because it can provide valuable insights into the underlying patterns and trends within the data.

For our study, we analyze Bitcoin closing price data from February 2015 to January 2023, as shown in figure 1a. It is then decomposed into its components, as shown in figure 1b. In both figures, the x-axes denotes daily dates.

As shown in figure 1b, the trend component exhibits a mixture of upward and downward movements with a large portion of the data is explained by the trends. Meanwhile, the residual component did not follow any discernible pattern. However, there was clear seasonality in the data. This seasonality occurs only within a range of 200 to -200, while the original price ranges around 60K. Therefore, even though there is a definite pattern, being significantly small, it will not be useful for actual prediction. This also suggests that the movement in Bitcoin prices is heavily influenced by external factors.

We explore two external factors, interest rates [3] and recession probability [4], to determine whether they have a positive impact on predicting Bitcoin prices. Figure 2a shows the recession probability data for February 2015 to January 2023. To better understand the correlation visually, we superimpose the data on top of Bitcoin with two y axes, as shown in figure 2b. In both figures, the x-axes denotes daily dates.

Figure 2b shows that the significant surge in Bitcoin's value in 2021 started immediately after the 100% recession probability period in March and April 2020. Similarly, superimposing the interest rate data on Bitcoin with two y-axes, as shown in figure 3, we observe that the recent fall in Bitcoin's price since March 2022 could be explained by the increase in interest rates in recent years.

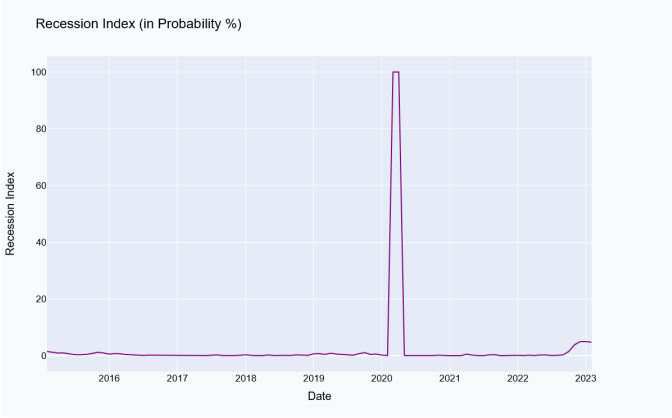
Based on these observations, we aim to explore these external factors in machine-learning models and check whether they have the potential to improve the accuracy of Bitcoin price predictions.

B. DATA PREPROCESSING

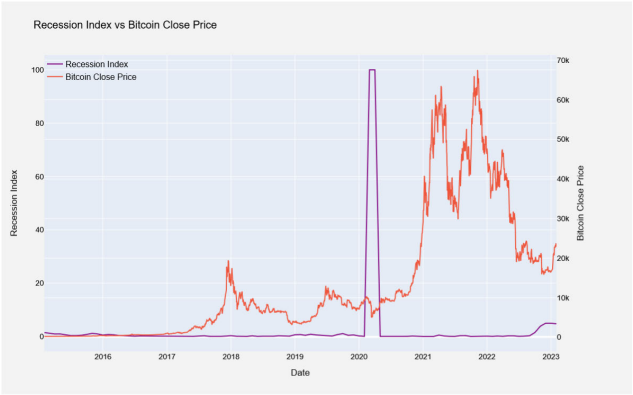
Data preprocessing is an essential step in any data analysis or modeling task because it is the foundation for generating accurate and reliable results. The data preprocessing process involves cleaning, transforming, and organizing to make it suitable for analysis and modeling. In our work, we use Bitcoin closing prices as our training and target variables [19]. As external factors, we use recession probability [4] and interest rate [3] as exogenous variables. Both exogenous variables are represented as percentages on a scale of 0 to 100. We downloaded all data within a date range of February 1, 2015, and February 1, 2023.

Bitcoin prices and interest rate data were collected on a daily basis, while recession probability data were available on a monthly basis. Therefore, we convert the recession probability data into daily frequency to match the frequencies of the other two variables. The total number of data points after preprocessing was 2922.

Our goal is to predict Bitcoin prices one month in advance. We hypothesize that the evaluative test dataset set should



(a) Recession probability from February 2015 to January 2023, measured as a percentage. The probability remained low for much of the period but shows a significant spike in March and April 2020, coinciding with the onset of the COVID-19 pandemic, reflecting the sudden economic downturn during that period.



(b) Comparative view of the Recession probability and Bitcoin closing prices from February 2015 to January 2023. This dual-axis chart overlays the Recession probability (in purple) with Bitcoin's closing price (in red), allowing for a visual examination of their relationship. A notable observation is the marked increase in Bitcoin's price immediately following the peak recession probability in March-April 2020, which may indicate a trend in investment behavior during economic downturns.

FIGURE 2. Recession probability and its comparison with Bitcoin price dynamics.

FIGURE 3. Overlay of interest rates and Bitcoin closing prices from February 2015 to January 2023. This visualization aligns the changes in interest rates (in blue) with Bitcoin's closing price (in red) on a dual-axis chart. The observable decline in Bitcoin prices since March 2022 coincides with a period of rising interest rates.

ideally be as large as the maximum forecast horizon we plan to use. To achieve this, we set aside one month for both validation and testing. Therefore, we set the training data from February 1, 2015, to November 30, 2022, and the validation data for December 2022, and the test data set for January 2023. Both the validation and test data consisted of 31 days. This arrangement helps the training phase to have as much data as feasible under our one month prediction horizon.

We check the data for null values to ensure the accuracy and reliability of our results. Fortunately, no null points were observed. However, we notice that the data contains commas, which we remove, and set all the data types as floats (by utilizing the astype function call of the pandas dataframe library). This ensures that the data are consistent and ready for further analyses.

C. LSTM MODEL

In this section, we discuss the different phases of the LSTM model chronologically. We plan to perform the task first using forecasting with no exogenous variables. We then analyze the impact of the recession probability followed by the interest rate separately. Finally, we combine all the data and examine the model outcomes.

1) LSTM MODEL WITH ONLY BITCOIN DATA

Initially, we construct a stacked LSTM model using only Bitcoin data as input. The model was designed to predict one day ahead to facilitate extrapolation. We then use Keras Tuner, an open-source Python library, to optimize the hyperparameters of the model [17]. Specifically, we use Keras Tuner to search for the optimal number of LSTM stacks with neurons and the number of neurons in the dense layer. The search spaces are listed in table 1. By tuning the hyperparameters of the model, we aim to find the best combination of layers and training parameters to achieve the lowest possible RMSE and MAPE.

TABLE 1. Finding best fit hyperparameter for an LSTM model with only Bitcoin data.

Name	Search Spaces
LSTM Stacks	1 to 4
LSTM neurons	50, 250, 500
Dense neurons	10 to 100 (step size 5)

We utilize the **TimeseriesGenerator** class from the Keras preprocessing sequence library to generate temporal batches of the training and validation sets. The training set consisted of data from February 1, 2015, to November 30, 2022, while the validation set consisted the month of December 2022. We use MinMaxScaler from the Python sklearn preprocessing library to normalize the data.

However, because the RMSE during the fitting process is only for one day ahead, we develop a separate validation script that allows us to evaluate the model's performance over longer time horizons. In our case, the duration was 31 days in December. This validation script follow the non-teacher forcing approach and output the RMSE for 31 days. By building and validating our stacked LSTM model using only Bitcoin data, we establish a baseline for subsequent experiments that will involve additional exogenous variables.

2) LSTM MODEL WITH RECESSION PROBABILITY

Here we explore the impact of recession probability as an exogenous variable using Bitcoin data. For the LSTM model, which incorporates both Bitcoin and recession probability data, we must ensure that the recession probability data are as fine-grained as the Bitcoin data. To achieve this, we write a script that spreads the monthly recession data into daily data points, which allows us to align the length of the recession data with the Bitcoin data for training and validation.

The recession data input is fed in parallel with Bitcoin. We design the model to accept two inputs: Bitcoin data and exogenous recession probability data. This allows us to see how the presence of recession probability affects the model's predictions. We use the non-teacher forcing validation script to accommodate the two inputs. This script generates a data frame of predicted values for December, which we then use to calculate the RMSE and MAPE.

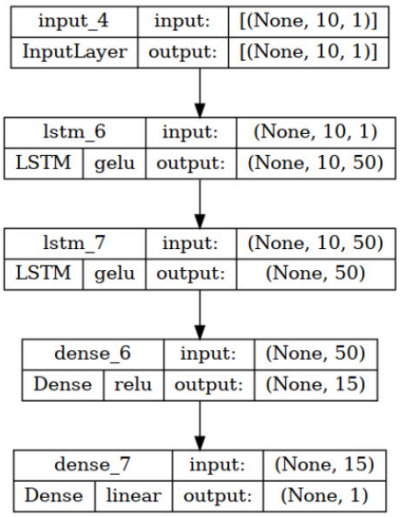


FIGURE 4. Architecture of the LSTM-based neural network used for Bitcoin price prediction. The model consists of two stacked LSTM layers with 50 neurons each, where the first layer is configured to return sequences to the next. A Gelu activation function is applied to all hidden layers, leading to a dense output layer with 15 neurons. Optimization is performed using the Adam optimizer, with mean squared error (MSE) as the loss function and Root Mean Squared Error (RMSE) for performance evaluation. The model comprises a total of 31,381 trainable parameters.

To calculate the RMSE, we utilize the Python statsmodel library, which provides a straightforward way to measure the difference between the predicted and actual values. Additionally, we create a custom MAPE function that calculates the percentage difference between the predicted and actual values, providing a sense of the accuracy of the model's predictions.

Initially, we normalize the recession probability data in the same way as we do for Bitcoin data using MinMaxScaler. However, after viewing the prediction graphs, we found that fluctuations in the prediction are relatively large. Therefore, we decide to change the normalization technique to a StandardScaler and attach a layer normalization after the recession input in the model. This significantly improves the performance of the model, and we obtain more accurate predictions.

3) LSTM MODEL WITH INTEREST RATE

Next, we explore the impact of interest rates as an exogenous variable and visualize its impact. In this case, interest rate data are already available as daily data, which means that we do not have to spread them out, as we did with recession probability data. Once we have the interest rate data, we create a script to prepare the training and validation data for both Bitcoin and the interest rate data, ensuring that they are of the same input length.

We then build the LSTM model to accept two inputs, Bitcoin data and exogenous interest rate data. We utilize the same non-teacher forcing validation script used for the recession probability model. The validation script produces a data frame of predicted values for December, which we then used to calculate both the RMSE and MAPE.

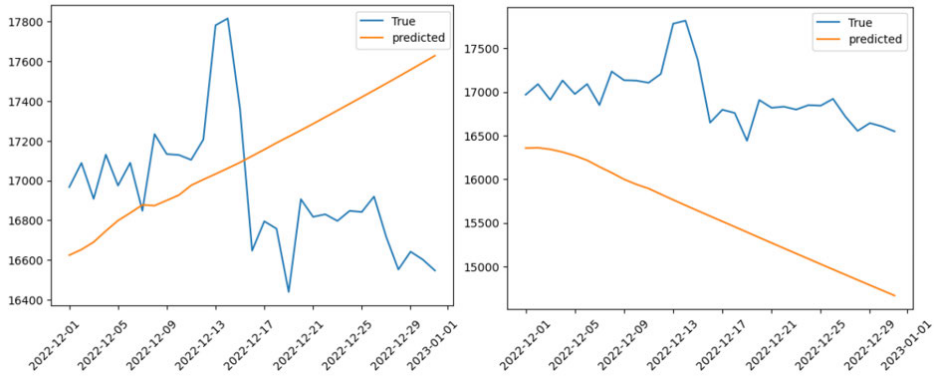


FIGURE 5. Two outcome scenarios from the LSTM predictive model using only Bitcoin data. The model's average prediction line displays minimal fluctuation, at times trending either upward or downward. This lack of variability in predictions is attributed to the model's limited information, which consists solely of Bitcoin-related information. The figure illustrates the challenges in capturing the inherent volatility of Bitcoin prices when using a uni-variate approach for forecasting, highlighting the potential risks of relying on Bitcoin data alone for prediction.

However, here we also run into some issues with the normalization technique for the interest rate data. Initially, we use MinMaxScaler to normalize the data, which results in an unstable loss curve. To address this, we switch to StandardScaler, followed by layer normalization after the interest rate input in the model. This improves the stability of the loss curve and results in more accurate predictions.

4) LSTM MODEL WITH RECESSION PROBABILITY AND INTEREST RATE

Finally, as we are clear, how the model respond to each of the data, we then apply both the exogenous variables as interest rate and recession probability using the Bitcoin data. In this case, we use the same data normalization methods as before, applying StandardScaler to the interest rate and recession probability and MinMaxScaler to Bitcoin.

To create the training and validation data, we utilize the same script as before to match the input length of Bitcoin. The exogenous variables are then fed into the model in parallel with the Bitcoin. In this phase, we design the model to accept three inputs: Bitcoin, the interest rate, and recession probability. We utilize Keras Tuner to finalize the number of dense layers and neurons before the final layer. We optimize the model performance by utilizing the Keras Tuner to fine tune both the model layers and hyperparameters.

To validate the model, we used the same non-teacher forcing validation script as before, with slight modifications to accept the three inputs. The validation script creates a data frame of the predicted values for December. We use the RMSE function from the Python statsmodel library to calculate the RMSE and utilize the custom made MAPE function to calculate the MAPE.

5) FINE TUNING

During the training of the LSTM model with all three exogenous variables, we notice that the performance of the

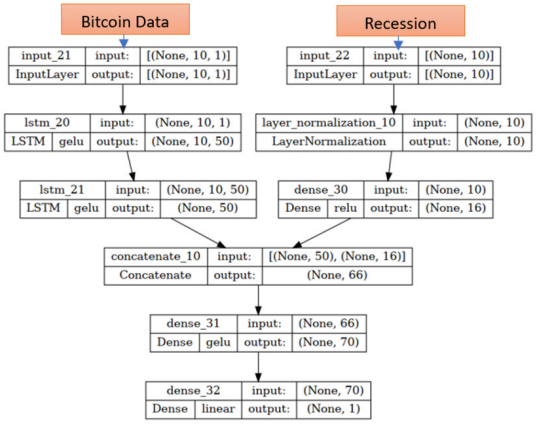
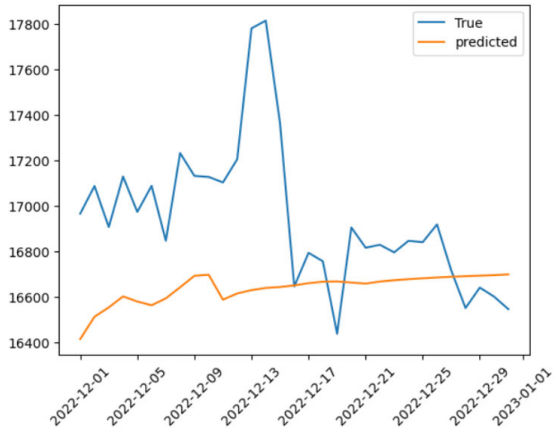


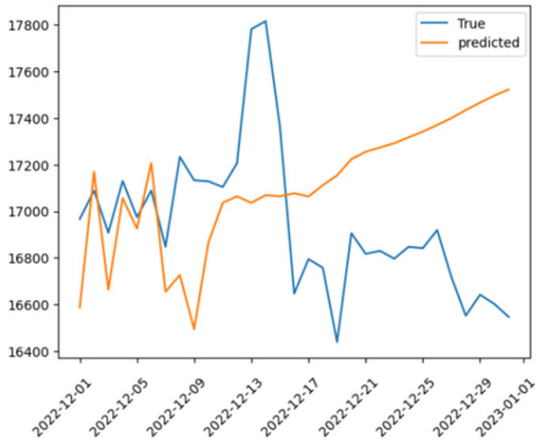
FIGURE 6. Enhanced LSTM model structure incorporating Bitcoin and recession probability data. The architecture retains the two stacked LSTM layers with 50 neurons, as established in the previous setup. In this iteration, the model processes recession probability data in parallel with Bitcoin data. After the initial processing, the outputs are merged and fed into a larger dense layer of 70 neurons. This augmented model comprises 37,341 trainable parameters and utilizes a refined learning rate of 0.00006, aiming to improve the robustness of Bitcoin price predictions by integrating economic indicators.

model was slightly inconsistent. Approximately three out of ten runs results in higher readings for both RMSE and MAPE. Further analysis reveal that the longer the model trains, the more it misses fluctuations and becomes prone to providing an average prediction line. This means that the fluctuations are retained during the early phase of training and then slowly disappears as the model continues to train.

To address this issue, we implemented EarlyStopping from the keras library. The functionality of EarlyStopping is to stop the training process early if certain criteria are met, such as if the validation RMSE loss does not decrease within a certain number of epochs. This helps us stop training early and prevents the model from overfitting to the training data. Additionally, EarlyStopping switches the weights back to the point where it achieves the first minimum



(a) Prediction output utilizing MinMaxScaler normalization on recession probability data with a ReLU activation function. This approach results in fluctuations in the predictive curve, contrasting with the average prediction line observed when using only Bitcoin data. .



(b) Adjusted prediction output after applying StandardScaler normalization to recession probability data. This modification leads to a more accurate representation of fluctuations, particularly noticeable at the beginning of the month. However, the model tends to average out predictions towards the latter half of the month, suggesting need for more information.

FIGURE 7. Fluctuations captured in first half of the month by adding recession probability to the model.

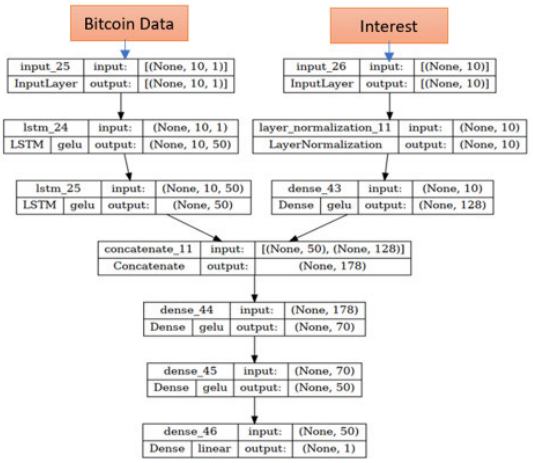


FIGURE 8. LSTM network model processing Bitcoin and interest rate data. The model follows the previous LSTM layers but adds a parallel input for interest rates, leading to two subsequent dense layers of 70 and 50 neurons. It has 48,159 trainable parameters, uses Gelu activation, and a learning rate of 0.0001.

convergence, ensuring that the model does not miss any important fluctuations during the training process.

Although we implement EarlyStopping to improve the consistency of our model, we notice that in some runs, the loss increases as early as from the second epoch. This leads us to consider the initial weights assigned by Keras, which may start the model in a poor position in the gradient descent process without any local minima around. To address this, we save the initial weights of a successful run and ensure that the model always starts from a specific region of gradient descent. By doing so, the model consistently begins from a point where there are enough local minima available, resulting in an improved performance.

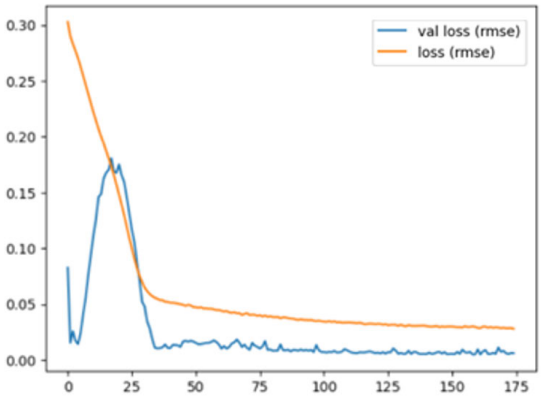
D. FACEBOOK PROPHET MODEL

We also explore the Prophet model, a well-known forecasting model, to predict Bitcoin using both exogenous variables. Although many of this model's parameters are tuned automatically, we take a closer look at some key hyperparameters to further optimize our results. Specifically, we tune the changepoint_prior_scale, seasonality_prior_scale, and n_changepoints parameters. The changepoint_prior_scale, which is similar to an L1 penalty, is known as the lasso penalty. On the other hand, the seasonality_prior_scale is more like an L2 penalty. Finally, the n_changepoints parameter is a crucial value that can impact the model's tendency to overfit or underfit. Setting it too high can lead to overfitting, whereas setting it too low can lead to underfitting.

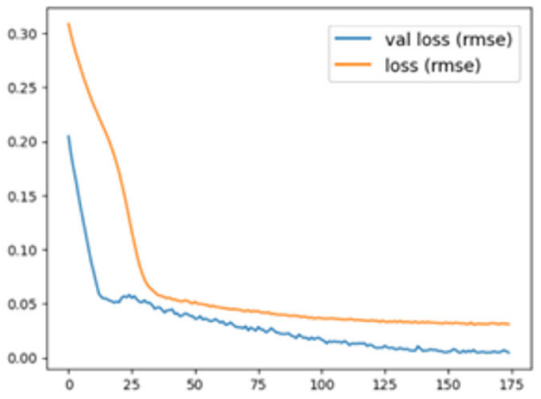
In our experiment, we pass all the data, including the two exogenous variables (interest rate and recession probability) with Bitcoin, training from February 2015 to November 2022 and predicting December. As the Facebook prophet automatically normalizes the data, we first pass the raw data to the model. However, we also check the results when we normalize the data in the same way as we did for LSTM, using StandardScaler for interest rate and recession probability, and MinMaxScaler for Bitcoin. For both cases, we follow the non-teacher forcing technique to calculate RMSE and MAPE.

E. MODEL EVALUATION

In this study, we design our models to predict one day at a time and extrapolate them to predict 31 days ahead. Extrapolation is a technique that extends the model's predictions by any range, and we prefer this in our case because it allows for more flexibility. This technique enables us to use the



(a) Training loss curve showing an abnormal spike in the initial epochs when MinMaxScaler is used for normalization, with loss peaking significantly before stabilizing.



(b) Loss curve demonstrating stabilization when StandardScaler and layer normalization techniques are applied. This contrasts with the earlier volatility observed with MinMaxScaler, indicating improved model training dynamics.

FIGURE 9. Addressing loss curve abnormality after adding interest data to the model.

same model architecture to predict different forecast horizons without having to train a new model for each specific forecast length.

We evaluate our models using a non-teacher forcing approach. The model learns from its own errors and handles uncertainty better. Non-teacher forcing is also known as true prediction. This is because, it brings real errors upfront when there are no validation data. This informs us of what the real case would be when we predict the unknowns.

All code for the methods above can be found online in the following GitHub repository: https://github.com/am2fq/BitcoinForecasting [20].

IV. RESULTS

A. LSTM MODEL

1) LSTM MODEL WITH ONLY BITCOIN DATA

We begin by analyzing the effect of relying only on Bitcoin without any additional regressors. As shown in figure 4, we utilize two stacked LSTM layers, each with 50 neurons, with the first LSTM layer having return sequences set to true. The reason behind this is, for each time step, the first LSTM layer need to provide an input to the following stacked LSTM layer. We implement all hidden layers with a Gelu activation function. We connect the stacked layers to a dense layer with 15 neurons. We utilize the Adam optimizer with the default learning rate. We choose 'mse' as loss and 'RootMeanSquaredError' as the metric. The total number of trainable parameters of the model is 31,381.

To determine the number of days for input data, we experiment with different ranges of 5 to 15 days, and finally decide to use 10 days of Bitcoin as the input to the model. We find that using 10 days provides a comparatively stable performance compared to the other ranges. Table 2 shows the RMSE and MAPE of 5, 10, and 15 days time-step input for the model.

TABLE 2. RMSE and MAPE for different time periods.

Timesteps	RMSE	MAPE
5 days	2979.5	14.4%
10 days	2500	12%
15 days	2577.5	14%

The best result we get is taking 10 days as input that results in an RMSE and MAPE of 2500 and 12% respectively as shown in table 2. We show two predictions of the model in figure 5. As can be seen, the model provides an average prediction line, which can be sometimes upward or downward capturing no fluctuations. This is because the model is confused, and we believe that this confusion is due to the limited data of only Bitcoin. In summary, while the LSTM model utilizing only Bitcoin data might hold potential for short-term forecasting, figure 5 shows that Bitcoin prices experience significant volatility over time. Consequently, we demonstrate that depending exclusively on Bitcoin data for predictive analysis could be precarious due to this inherent instability.

2) LSTM MODEL WITH RECESSION PROBABILITY

We now discuss the effect of training the LSTM model with both Bitcoin and recession probability data. First, we convert the monthly recession probability data to daily data to match the frequency of the Bitcoin data. Because the model provides an output of a day ahead, we create the training and validation sequences by shifting one day step to the right.

Except for the dense layer of 15 neurons as shown in figure 6, the LSTM layer architecture is kept the same as in the previous section, where we used two stacked LSTM layers with 50 neurons. In this case, we feed the recession probability data alongside Bitcoin data in parallel. Following a dense layer of 16 neurons, we concatenate the output from the LSTM and recession probability and pass in through a

FIGURE 10. Prediction graph comparing actual values ('True') and model outputs ('predicted'). The model, adjusted with interest rate data, effectively captures fluctuations in the latter half of the month, while the first half shows a generalized average trend.

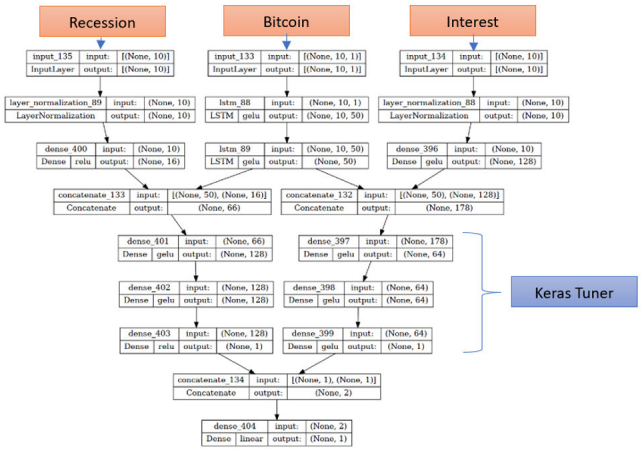


FIGURE 11. Detailed LSTM model architecture integrating Bitcoin data with recession probability and interest rate inputs. The structure above the concatenation layers is consistent with previous models, with Keras Tuner optimizing the network's configuration. A final concatenation layer merges outputs before passing them to the concluding dense layer. This model has 73,125 trainable parameters, with a learning rate fine-tuned to 0.00007.

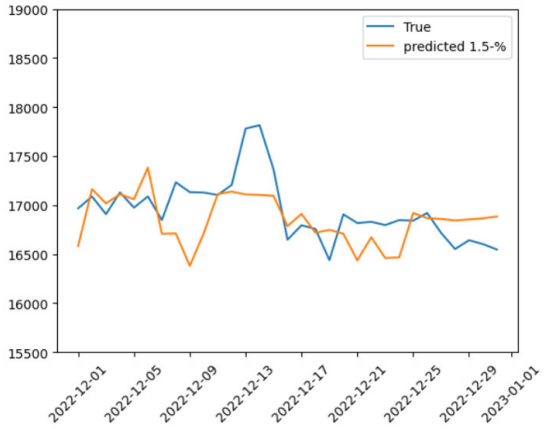


FIGURE 12. Model predictions versus actual Bitcoin prices for December 2022, incorporating both interest rate and recession probability data. This approach preserves the fluctuations throughout the month, reflecting in a low mean absolute percentage error (MAPE) of 1.5%.

dense layer of 70 neurons. The overall architecture now have a total of 37,341 trainable parameters. In this case, we use a learning rate of.00006.

We use the ReLU activation function for recession probability, as it is provides a better convergence than the Gelu activation function. Initially, we normalize the data using MinMaxScaler from the Python sklearn library. However, we notice that instead of having an average prediction line, as in figure 5, fluctuations occurred in the output. This can be observed in figure 7a.

It can be noticed that the fluctuations were far too off. To address this issue, we change the data normalization method to StandardScaler, resulting in much better fluctuations in the output. Figure 7b shows the improved output with the StandardScaler. We notice that by adding recession probability, the model was picking up fluctuations in the 1st of the month, but is giving an average prediction for the latter half.

3) LSTM MODEL WITH INTEREST RATE

Interest rate data were already in daily format, so there is no need to re-sample the data. We create the training and validation sets in the same way, making a vector of 10 time-steps and shifting by one day for the next set.

The LSTM layer is kept the same as in the previous section except the dense layer of 15 neurons as shown in figure 8. We feed the interest rate in parallel with the Bitcoin data. Following a dense layer of 128 neurons, we concatenate the output from LSTM and the interest rate and pass in through two hidden dense layers. These hidden dense layers are now having 70 and 50 neurons respectively. The model now have a total of 48,159 trainable parameters. In this case, we use a learning rate of .0001. Unlike recession probability, we use Gelu activation function.

In this case, both normalization techniques, MinMaxScaler and StandardScaler, can capture fluctuations in the data. However, an abnormality is observed in the loss curve when MinMaxScaler is applied. The loss increases significantly for approximately 13 epochs, as shown in figure 9a. The issue is resolved by applying StandardScaler with layer normalization, as shown in figure 9b. From the prediction graph in figure 10, the model now captures fluctuations in the latter half of the month but provides an average prediction line for the first half.

4) LSTM MODEL WITH RECESSION PROBABILITY AND INTEREST RATE

In the final LSTM model, we add both exogenous variables, interest rate and recession probability, in parallel with Bitcoin data to predict the price for December 2022. Once again, we convert the monthly recession probability data to daily data to match the length of all other data frames. We make the

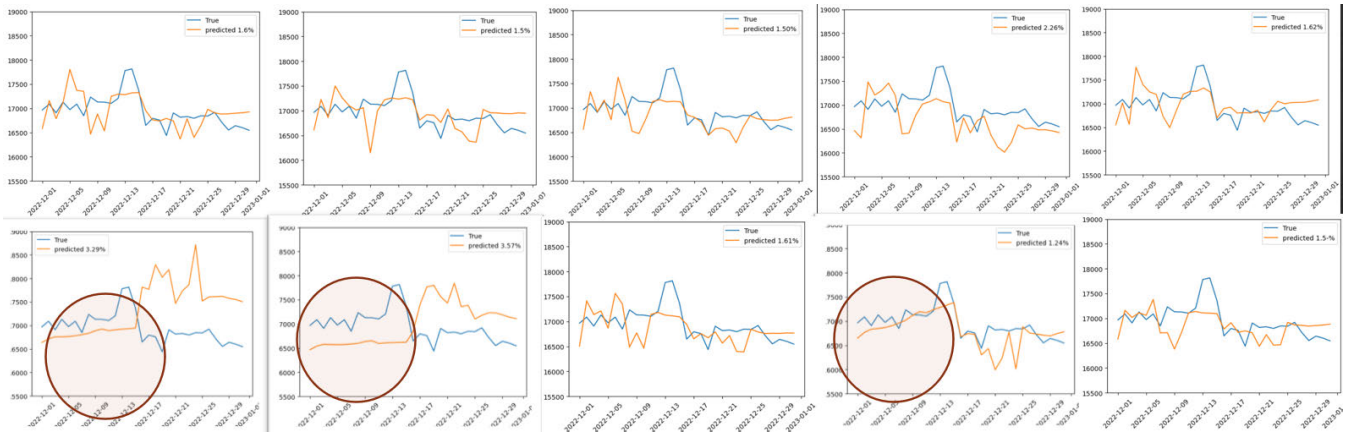


FIGURE 13. Ten consecutive runs of the LSTM model to assess prediction consistency for Bitcoin price, with highlighted sections indicating abnormal behavior. Despite generally preserving fluctuations, about three in ten iterations exhibit a higher MAPE, particularly in the first half of the month, underscoring areas for model improvement.

```
earlystop_callback = keras.callbacks.EarlyStopping(
    monitor='val_root_mean_squared_error',
    min_delta=0.0,
    patience=13,
    verbose=1,
    mode='min',
    baseline=None,
    restore_best_weights=True
)
epoch = 90
history = model.fit([x_train,exo_interest_train,exo_recession_train],y_train,
                    validation_data=([x_val,exo_interest_val, exo_recession_val],y_val),
                    epochs = epoch)
```

FIGURE 14. EarlyStopping hyper-parameters that worked the best to stop from training any further.

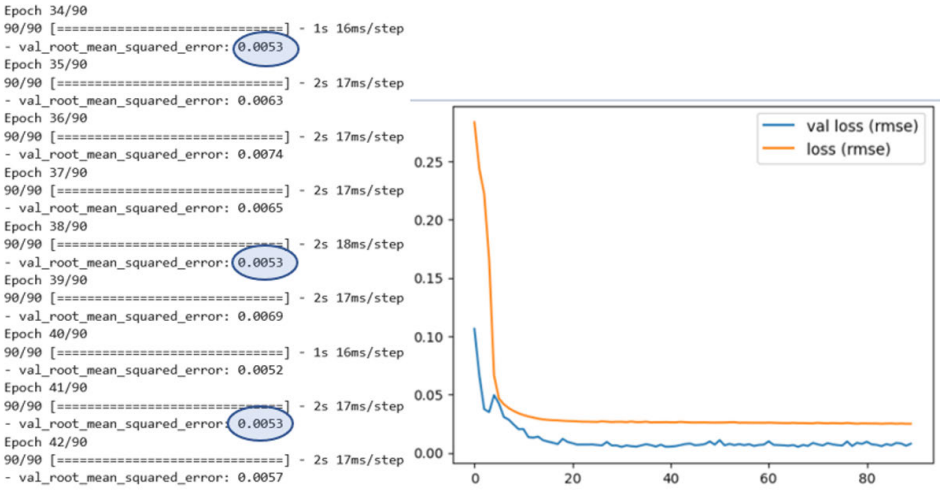


FIGURE 15. Training performance over 90 epochs, detailing instances where validation loss reaches 0.0053 multiple times (left), alongside the corresponding loss curve (right). The loss stays low, but the model increasingly favors a smooth average prediction by forgetting true data fluctuations as training progresses, implying a need for careful epoch selection.

training and validation sequences by sliding a day step to the right. We use the same normalization techniques as discussed in the previous sections.

The architecture above the concatenation layer remains the same as shown in figure 11, resulting in two concatenation layers: one for the interest rate and the other for the recession

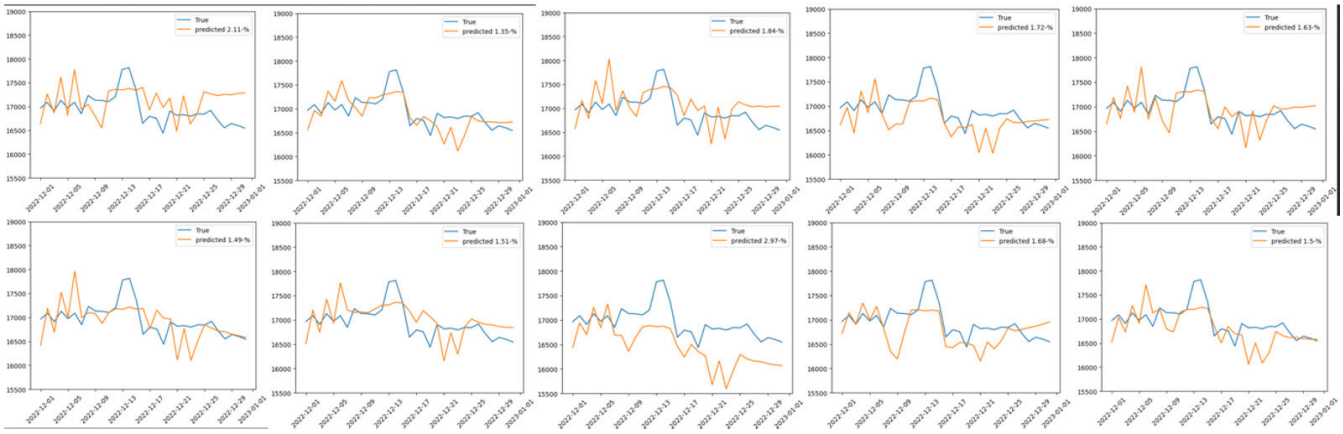
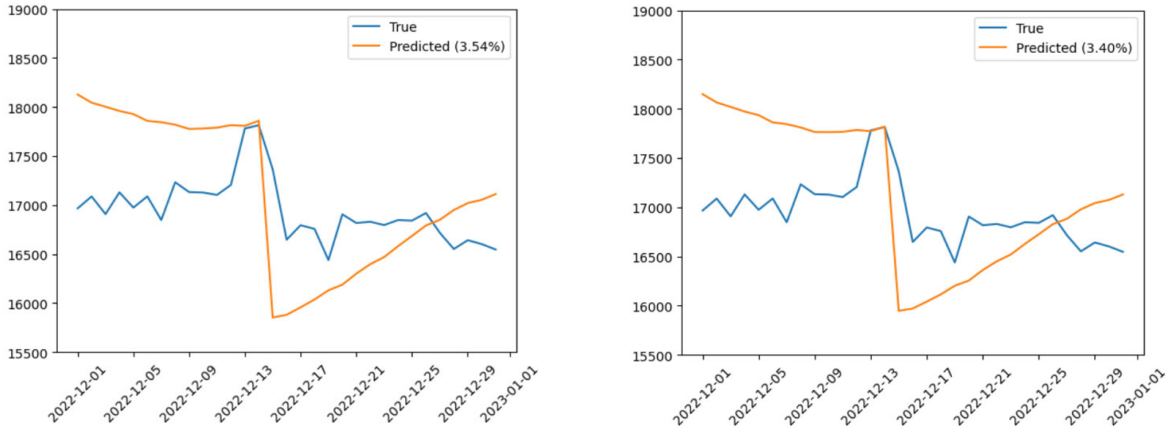


FIGURE 16. Ten subsequent runs of the LSTM model after fine-tuning, showing consistent capture of fluctuations across different iterations, with a stable performance and prediction accuracy in each case.



(a) Prediction of December 2022 by Prophet model. (Data not normalized).

(b) Prediction of December 2022 by Prophet model. (Data normalized).

FIGURE 17. Visualizing data normalization effect on Prophet model.

probability. We subject the former part of the concatenation layer to Keras Tuner to find the best number of hidden layers and neurons. We add a concatenation layer that combines the processed tensors from both parallel branches before the final dense layer. The total number of trainable parameters is 73,125. However, in this case we lower the learning rate to .00007.

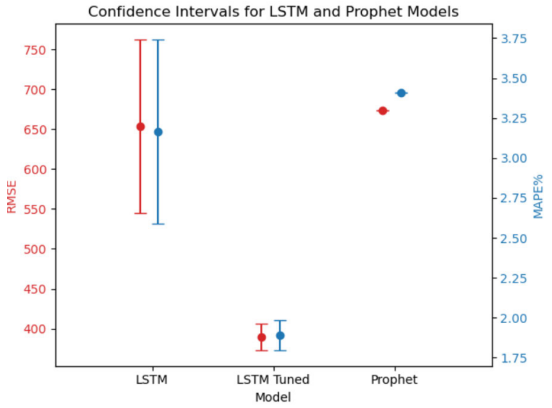
As expected, the model captures mostly all the fluctuations for December 2022, as shown in figure 12, and the combination of all the exogenous variables improved the accuracy of the prediction compared to the individual models, giving an MAPE of 1.5%.

5) FINE TUNING

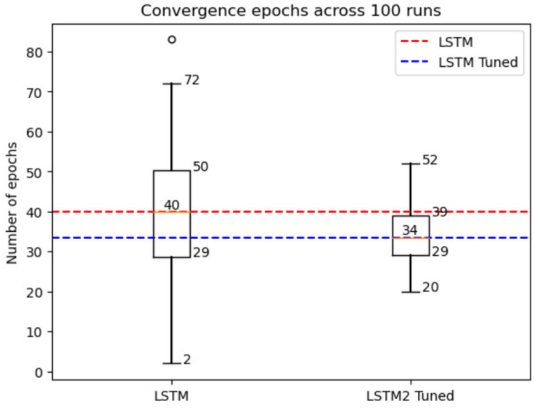
Fine-tuning a machine learning model is a crucial step in ensuring that it performs consistently and accurately. In the case of our LSTM model for Bitcoin price prediction, we recognize the importance of evaluating consistency. We run the model multiple times and notice that approximately three out of ten runs result in a high MAPE as shown in figure 13, providing an average prediction line for the first half of the month. This inconsistency is not desirable, and we aim to identify and address the underlying causes.

One possible reason for this inconsistency is the length of the training process. When the model trains for too long, it is more likely to converge to an average prediction line, thereby losing its ability to capture fluctuations in the data. We can visualize this by examining the sample training curve for 90 epochs, as shown in figure 15, which shows that the loss drops below a certain value multiple times during training. To address this issue, we implement the EarlyStopping method from Keras. We set a patience level of 13, as shown in figure 14.

However, even with EarlyStopping, we observe that the loss for some runs keeps rising from the very second epoch, indicating a bad start in the gradient descent. To overcome this issue, we save the initial weights of a successful run and load



(a) Confidence intervals for LSTM and Prophet models, with RMSE on the left y-axis (red) and MAPE on the right y-axis (blue). The 'LSTM' denotes the initial model, and 'LSTM Tuned' indicates the model with saved weights, which shows notably narrower confidence intervals, suggesting improved reliability and precision in forecasting.



(b) Box plot displaying the median convergence epochs for LSTM models using early stopping. This visualization offers insight into the variability of convergence across multiple runs, highlighting the median epoch at which the models typically stabilize.

FIGURE 18. Statistics gathered over one hundred runs.

them before training each subsequent run. This ensures that the model always starts from a good point, thereby improving its consistency.

By tuning these issues, we observe that our LSTM model captures fluctuations in the data consistently, regardless of how many times it is run. This is shown in figure 16, where the model's performance is consistently good, indicating that the fine-tuning process successfully addressed the issues with model inconsistency.

B. FACEBOOK PROPHET MODEL

In addition to evaluating the performance of LSTM, we also explore the use of the Facebook Prophet model to predict the price of Bitcoin using two exogenous variables: recession probability and interest rate. To optimize the performance of the Prophet model, we conduct a random search of 1000 runs to find the best combination of three hyperparameters: changepoint_prior_scale, seasonality_prior_scale, and n_changepoints. The top five results are tabulated in table 3. Based on the results, we select the best performing set of hyperparameters.

TABLE 3. Top five best hyper-parameter combinations.

Serial	changepoint_prior_scale	seasonality_prior_scale	n_changepoints
1	0.458151242	8.81013758	9
2	0.47096539	1.294920767	9
3	0.411642353	5.344713499	9
4	0.439647171	0.715646787	9
5	0.383279271	6.640903921	9

We train the model on all the available data from February 2015 to November 2022, and use it to predict the month of December 2022. The predicted results are shown in figure 17a. None of the data were normalized, as the Prophet model takes care of it automatically. The resulting MAPE was

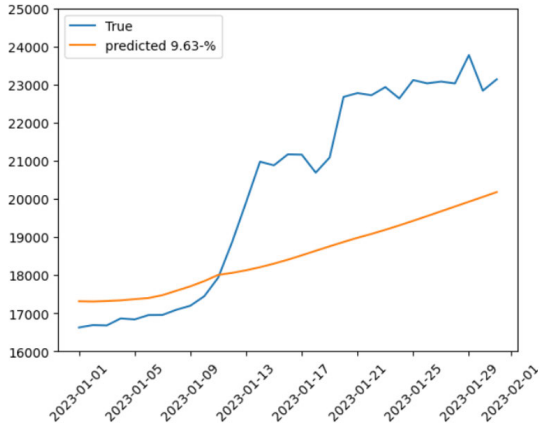
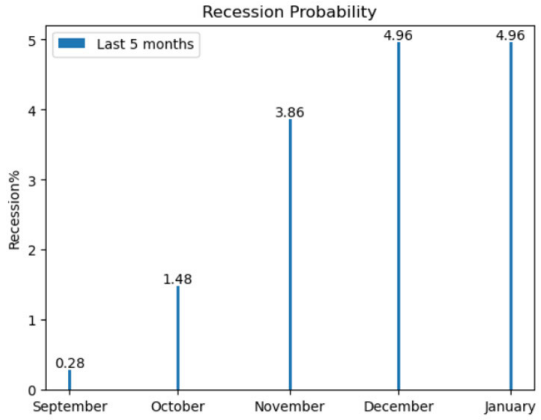


FIGURE 19. LSTM model's test data predictions, resulting in a MAPE of 9.63% and an RMSE of 2487.29. These increased error metrics is due to the absence of changepoints in the exogenous variables.

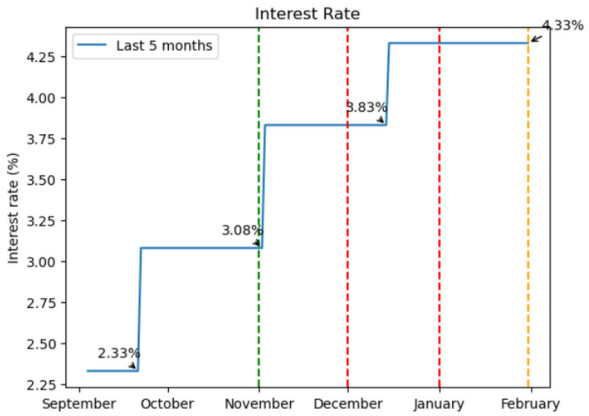
3.54%. However, we also check by normalizing the data in the same way as for the LSTM model. The normalized results are shown in figure 17b.

It is evident from figure 17b that the MAPE is slightly improved compared to that in figure 17a. However, both figures missed capturing most of the fluctuations in the actual price of Bitcoin, except for one drop in the middle of the month. It is also worth noting that the Facebook Prophet model is deterministic. Regardless of the number of runs, it results with the same RMSE and MAPE. This is also the reason why we notice no error bars in the confidence interval graph for the Prophet model in figure 18a.

Overall, our experiments with the Facebook prophet model did not yield results as promising as those obtained using the LSTM model. This was expected because the Bitcoin data are relatively unstable and highly volatile.



(a) Recession probability increased from 3.86% to 4.96% in December, but remained same for January.



(b) Interest rate increased from 3.83% to 4.33% in December, but remained same throughout January.

FIGURE 20. Visualization of recession probability and interest rates over the last five months showing stability in January 2023, with unchanged values from December. This uniformity suggests that the fluctuations in Bitcoin prices for January were influenced by factors other than the included exogenous variables, resulting in the model generating an average prediction line.

C. CONFIDENCE INTERVAL

To obtain confidence intervals, we run each model 100 times and gather the results. We then report the RMSE and MAPE of each run, resulting in two y-axes for each model.

Figure 18a, presents the confidence intervals for each model. The left y-axis is marked in red, representing the RMSE, while the right y-axis is marked in blue, representing the MAPE in percentage. The x-axis labels represent the models, where the label with ''LSTM'' denotes the model where the weights were not saved, while the ''Fine-tuned'' model refers to the model in which weights are saved. For each model, we present two graphs, one for the RMSE and another for the MAPE.

Our results demonstrate that the LSTM model with saved weights performs the best, with the lowest RMSE and MAPE. Figure 18a clearly shows that the confidence interval for the fine-tuned LSTM model is smaller than those for the other models.

However, owing to the early stopping method used in the LSTM models, each run of the model converged at slightly different epochs. Therefore, to further understand the convergence behavior, we present a median box plot for each model in figure 18b, which provides an overview of the convergence behavior of the LSTM models and shows the median converging epochs.

D. TEST DATA

The test data play a crucial role in evaluating the performance of a model. In this section, we discuss the prediction results on the test data, which is the month of January 2023. The hyperparameters for all three models are maintained the same as those during the evaluation of the validation data.

1) LSTM MODEL

The prediction results of the LSTM model on the test data are shown in figure 19. We observe that the MAPE and

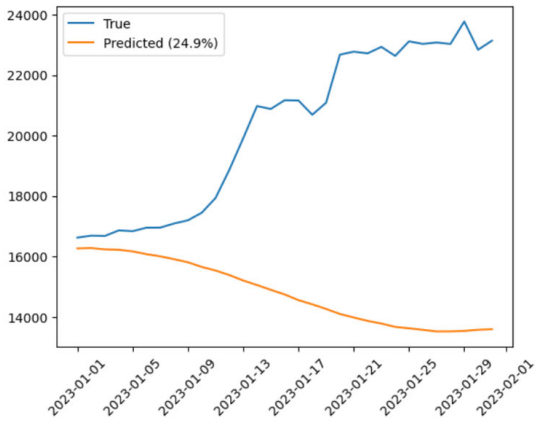


FIGURE 21. January 2023 predictions using the Prophet model, showing a high MAPE of 24.9% and an RMSE of 6545.15. These increased error measures, significantly higher than the validation data metrics, suggest that the model's performance was affected by the same lack of changepoints in January's exogenous variables, leading to an average prediction line.

RMSE increased to 9.63% and 2487.29 respectively. The relatively poor performance can be attributed to the absence of any changepoints in the exogenous variables, recession probability, and interest rate.

In figures 20a and 20b, we show the data for the last five months for the two exogenous variables. We notice that unlike December, January 2023 has no changepoint in recession probability. The recession probability in January was 4.96%, which is the same as that in December. Similarly, interest rates remained the same in January, and there is no change point.

These observations indicate that fluctuations in Bitcoin during January were not triggered by either of the exogenous variables. Therefore, the model provides an average prediction line.

In conclusion, the LSTM model's performance on the test data is not as good as that of the validation data because of

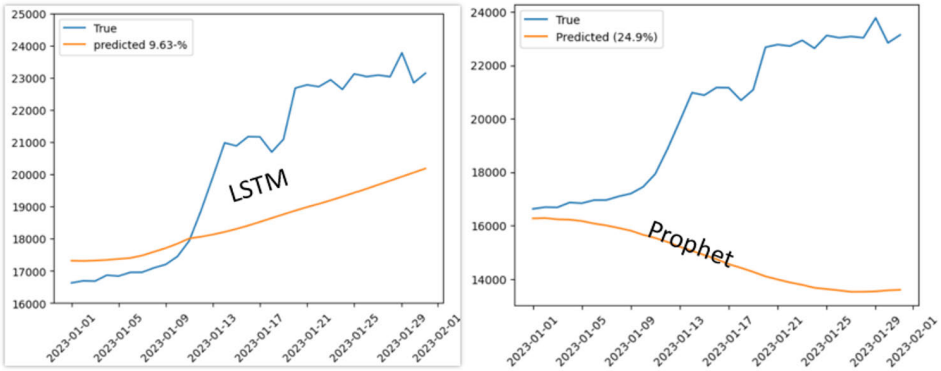


FIGURE 22. Comparative predictions for January 2023 with the LSTM (left) consistently indicating an upward trend and the Prophet model (right) showing a downward prediction. Even though the MAPE values are higher, the LSTM model consistently predicts an upward trend, indicating its ability to reliably capture the general direction of Bitcoin price movements.

the absence of any changepoints in the exogenous variables. However, this observation highlights the importance of selecting the relevant exogenous variables for accurate Bitcoin price predictions.

2) FACEBOOK PROPHET MODEL

In this subsection, we discuss the test data results for the Prophet model. As mentioned earlier, the test data used were for January 2023.

Figure 21 shows the prediction line for January 2023 using the prophet model. The MAPE value increased to 24.9%, which is quite high, and the RMSE value increased to 6545.15. These values are significantly higher than those obtained for the validation data.

The reasons for the poor performance of the Prophet model on the test data are the same as those discussed earlier. Specifically, there are no changepoints in the exogenous variables for January 2023, which means that the fluctuations in Bitcoin are not triggered by either of these variables. As a result, the model is giving an average prediction line.

However, it is worth noting that regardless of its poor accuracy, the LSTM model never failed to show an upward trend. This is particularly evident when we compare the results side by side in figure 22. No matter how many times we run the LSTM model, it never gave a downward prediction, as in the Prophet model. This also highlights the robustness of the LSTM model in predicting Bitcoin price.

V. DISCUSSIONS

In this project, we aimed to analyze the impact of exogenous variables, specifically interest rates and recession probability, on the prediction Bitcoin prices. We utilized two different models, LSTM and Prophet, to compare their performance in forecasting.

Our results indicate that consideration of exogenous variables is crucial for better forecasting. Without any outside variables, the models provided an average prediction line, demonstrating the importance of incorporating exogenous variables into the models.

We found that, given the changepoints, interest rates and recession probability have a significant impact on Bitcoin prices. By incorporating recession data with Bitcoin prices, the model was able to capture fluctuations in the first half of December. This was because there was a change in the recession at the beginning of the month. In the second half of December, the combination of interest rates and Bitcoin prices captured fluctuations, as there was a change point in the interest rate in the middle of the month. Combining both recession data and interest rates with Bitcoin prices resulted in capturing the fluctuations December.

When evaluating the models on the test data, we observed a poor performance for both models. This happened because there was no changepoint in either of the exogenous variables. Consequently, the models did not react and provided an average prediction line. However, regardless of the poor performance, the LSTM model was able to at-least inform the correct (increasing) trend of Bitcoin's price. This was not the case with the Facebook Prophet model, as we obtained an opposite (decreasing) trend with respect to the original data. To substantiate the superior performance of the LSTM model over the Facebook Prophet model, we employed several key performance indicators, including the RMSE and the MAPE. These metrics provide a quantitative basis for comparing the accuracy of the predictions made by each model. Additionally, we closely analyzed the nuanced fluctuations within the predictive versus actual price curves. This analysis enabled us to assess how well each model captures the inherent volatility of Bitcoin prices, further highlighting the LSTM model's enhanced ability to accurately reflect market dynamics.

In conclusion, we demonstrate that the interest rate and recession probability have a positive impact on forecasting the future prices of Bitcoin. Furthermore, to develop a

TABLE 4. Bitcoin closing price of last 20 days.

Dates	Price
2023-01-12	18869.58789
2023-01-13	19909.57422
2023-01-14	20976.29883
2023-01-15	20880.79883
2023-01-16	21169.63281
2023-01-17	21161.51953
2023-01-18	20688.78125
2023-01-19	21086.79297
2023-01-20	22676.55273
2023-01-21	22777.62500
2023-01-22	22720.41602
2023-01-23	22934.43164
2023-01-24	22636.46875
2023-01-25	23117.85938
2023-01-26	23032.77734
2023-01-27	23078.72852
2023-01-28	23031.08984
2023-01-29	23774.56641
2023-01-30	22840.13867
2023-01-31	23139.28000

TABLE 5. Recession probability of last 20 days.

Dates	Recession Probability
2023-01-12	4.96
2023-01-13	4.96
2023-01-14	4.96
2023-01-15	4.96
2023-01-16	4.96
2023-01-17	4.96
2023-01-18	4.96
2023-01-19	4.96
2023-01-20	4.96
2023-01-21	4.96
2023-01-22	4.96
2023-01-23	4.96
2023-01-24	4.96
2023-01-25	4.96
2023-01-26	4.96
2023-01-27	4.96
2023-01-28	4.96
2023-01-29	4.96
2023-01-30	4.96
2023-01-31	4.96

final model for accurate Bitcoin price forecasting, every month of a year should be tested to determine all possible impacting factors. As in our case, we have seen that the interest rate and recession probability have a significant impact on December, while this was not the case for January. However, we employed a strategy where we saved the weights from a successful run and initiated subsequent training from that point. Without this approach, we observed some training runs experiencing a rapid increase in loss as early as the second epoch. Our model's initial weights were set using Xavier (or Glorot) initialization. Future research could explore alternative weight initialization methods, such as He initialization or orthogonal initialization, to potentially enhance model performance. Additionally, incorporating these economic indicators alongside other variables, such as data from other cryptocurrencies, may further improve prediction accuracy.

TABLE 6. Interest rate of last 20 days.

Dates	Interest Rate
2023-01-12	4.33
2023-01-13	4.33
2023-01-14	4.33
2023-01-15	4.33
2023-01-16	4.33
2023-01-17	4.33
2023-01-18	4.33
2023-01-19	4.33
2023-01-20	4.33
2023-01-21	4.33
2023-01-22	4.33
2023-01-23	4.33
2023-01-24	4.33
2023-01-25	4.33
2023-01-26	4.33
2023-01-27	4.33
2023-01-28	4.33
2023-01-29	4.33
2023-01-30	4.33
2023-01-31	4.33

APPENDIX A
DATASET SNAPSHOTS

To have a clear idea of how the three datasets look, we add snapshots here.

A. BITCOIN CLOSING PRICE

The last 20 rows of the Bitcoin data frame are presented in table 4.

B. RECESSION PROBABILITY

The last 20 rows of the recession probability data frame is shown in table 5.

C. INTEREST RATE

The last 20 rows of the interest data frame is shown in table 6.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[2] S. J. Taylor and B. Letham, "Forecasting at scale," *Amer. Statistician*, vol. 72, no. 1, pp. 37–45, Jan. 2018, doi: 10.1080/00031305.2017.1380080.

[3] Board Governors Federal Reserve Syst. (U.S.). *Federal Funds Effective Rate*. Accessed: Feb. 5, 2023. [Online]. Available: https://fred.stlouisfed.org/series/DFF

[4] M. Chauvet and J. M. Piger. *Smoothed U.S. Recession Probabilities*. Accessed: Feb. 7, 2023. [Online]. Available: https://fred.stlouisfed.org/series/RECPROUSM156N

[5] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, no. 3, pp. 215–236, Apr. 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0925231295000399

[6] H. White, "Economic prediction using neural networks: The case of IBM daily stock returns," in *Proc. IEEE Int. Conf. Neural Netw.*, Jul. 1988, pp. 451–458.

[7] Y. Yoon and G. Swales, "Predicting stock price performance: A neural network approach," in *Proc. 24th Annu. Hawaii Int. Conf. Syst. Sci.*, 1991, pp. 156–162.

[8] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econ. Rev.*, vol. 29, nos. 5–6, pp. 594–621, Aug. 2010, doi: 10.1080/07474938.2010.481556.

[9] C. Chatfield and M. Yar, "Holt-winters forecasting: Some practical issues," *Statistician*, vol. 37, no. 2, p. 129, 1988. [Online]. Available: https://www.jstor.org/stable/2348687?origin=crossref
[10] A. Çibikdiken and E. Karakoyun, "Comparison of ARIMA time series model and LSTM deep learning algorithm for Bitcoin price forecasting," in *Proc. 13th Multidisciplinary Academic Conf. Prague*, 2018, p. 171.
[11] M. Cai, M. Pipattanasomporn, and S. Rahman, "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques," *Appl. Energy*, vol. 236, pp. 1078–1088, Feb. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261918318609
[12] I. Yenidogan, A. Çayir, O. Kozan, T. Dag, and Ç. Arslan, "Bitcoin forecasting using ARIMA and PROPHET," in *Proc. 3rd Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2018, pp. 621–624.
[13] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," 2017, *arXiv:1703.04691*.
[14] I. Georgoula, D. Pournarakis, C. Bilanakos, D. Sotiropoulos, and G. M. Giaglis. *Using Time-series and Sentiment Analysis to Detect the Determinants of Bitcoin Prices*. Accessed: Jan. 28, 2023. [Online]. Available: https://papers.ssrn.com/abstract=2607167
[15] S. McNally, J. Roche, and S. Caton, "Predicting the price of Bitcoin using machine learning," in *Proc. 26th Euromicro Int. Conf. Parallel, Distrib. Netw.-based Process. (PDP)*, Mar. 2018, pp. 339–343.
[16] Y. B. Kim, J. G. Kim, W. Kim, J. H. Im, T. H. Kim, S. J. Kang, and C. H. Kim, "Predicting fluctuations in cryptocurrency transactions based on user comments and replies," *PLoS ONE*, vol. 11, no. 8, Aug. 2016, Art. no. e0161197. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.016119
[17] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, and L. Invernizzi. (2019). *Kerastuner*. [Online]. Available: https://github.com/keras-team/keras-tuner
[18] *EarlyStopping*. Accessed: Feb. 1, 2023. [Online]. Available: https://keras.io/api/callbacks/early_stopping/
[19] *Bitcoin USD (BTC-USD) Price History & Historical Data—Yahoo Finance*. Accessed: Feb. 3, 2023. [Online]. Available: https://finance.yahoo.com/quote/BTC-USD/history/
[20] A. Mahfooz. (Apr. 2023). *BitcoinForecasting*. [Online]. Available: https://github.com/am2fq/BitcoinForecasting

ADEL MAHFOOZ received the B.S. degree in electrical and electronic engineering from the Ahsanullah University of Science and Technology, Bangladesh, in 2013. He is currently pursuing the Master of Science degree in computer science with Middle Tennessee State University (MTSU).

From 2013 to 2020, he was a Telecommunications Engineer with Telenor Grameenphone. There he devised automation solutions streamlining sensitive tasks to reduce errors and performed predictive fault analysis. His research interests include deep learning, forecasting, and cloud computing. He is a member of The Honor Society of Phi Kappa Phi. In 2023, he won the Hackers Choice Award from the Hackathon event held by MTSU.

JOSHUA L. PHILLIPS received the B.S. degree in computer science from Middle Tennessee State University, in 2002, the M.S. degree in computer science from Vanderbilt University, in 2004, and the Ph.D. degree in electrical engineering and computer science from the University of California, Merced, in 2012.

He is currently an Associate Professor of computer science and a Faculty Affiliate with the Data Science Program and Computational and Data Science Ph.D. Program, Middle Tennessee State University, Murfreesboro, TN, USA. Some recent publications include "Attention is not Enough" in Proceedings of the 44th Annual Meeting of the Cognitive Science Society (CogSci 2022), Toronto, Canada, "A Neurobiologically-Inspired Deep Learning Framework for Autonomous Context Learning" in Proceedings of the 33rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2021), and "Transfer Reinforcement Learning Using Output-Gated Working Memory" in Proceedings of the 34th AAAI Conference on Artificial Intelligence.