Bitcoin Price Prediction Using Machine Learning

By

Tang Jian Yang

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

May 2023

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Bitcoin Price Prediction Using Machine Learning

Academic Session: 202305

I TANG JIAN YANG

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:

1321, Jalan Emas 2,

Taman Bandar Baru,

31900, Kampar, Perak.

Tong Dong Ling

Supervisor's name

Date: 15/9/2023

Date: 15 Sep 2023

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY/INSTITUTE* OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 15/9/2023

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that Tang Jian Yang (ID No: 20ACB05721) has completed this final year project/ dissertation/ thesis* entitled "Bitcoin Price Prediction Using Machine Learning" under the supervision of Dr Tong Dong Ling (Supervisor) from the Department of Computer Science, Faculty/Institute* of Facuty of Information and Communication Technology, University Tunku Abdul Rahman.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

Tang Jian Yang

DECLARATION OF ORIGINALITY

I declare that this report entitled "**METHODOLOGY, CONCEPT AND DESIGN OF A 2-MICRON CMOS DIGITAL BASED TEACHING CHIP USING FULL-CUSTOM DESIGN STYLE**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature	:	___________________________
Name	:	Tang Jian Yang
Date	:	14/9/2023

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Dr Tong Dong Ling and my moderator, Dr Lim Jia Qi who have given me this bright opportunity to engage in a research project. It is my first step in machine learning research. A million thanks to you.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

Bitcoin price prediction is the act of forecasting future value of Bitcoin. A successful prediction of Bitcoin future value will maximize investor’s gains. Over the past few years, Bitcoin has been a topic of many, from investors to researchers, even ordinary citizens. Bitcoin is the first, largest and most valuable cryptocurrency till today. However, Bitcoin’s nature is very volatile and highly fluctuate which makes investing in Bitcoin feels like gambling to investors. It is very risky to invest in it as its price go up and down a lot within 1 day interval. Numerous studies have conducted on Bitcoin price prediction using traditional time series forecasting algorithms. In recent years, researchers have started using deep learning models to predict Bitcoin price as well. This study proposes three types of machine learning algorithms (LSTM, GRU, and Prophet) with two types of architectural configurations (Sequence-to-Sequence and Sequence-to-One) to predict Bitcoin’s closing price based on 1 year of Bitcoin historical data, (2, April 2022 to 2, April 2023). The data is split into 335 days for training set and 30 days for testing set. Three experiment was conducted (Sequence-to-Sequence Walk Forward, Sequence-to-One Walk Forward, and Sequence-to-One Rolling Origin). The performance of the proposed models is evaluated using Bitcoin price from 3, March, 2023 to 1, April 2023. The results on the models using various evaluation metrics such as RMSE, MAPE and MAE show that LSTM is the optimal model compared to GRU and Prophet. GRU is a second close. However, Prophet struggles to predict fluctuations and curvature in this highly unstable Bitcoin price prediction task. Having lower error metrics does not imply that the model is good.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AR	Autoregressive
MA	Moving Averages
ARIMA	Autoregressive Integrated Moving Average
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
MAE	Mean Absolute Error

CHAPTER 1

In this chapter, the introduction, motivation, problem statement, research objectives and project scope of the thesis are presented. The contribution to the field and the outline of the thesis is to determine the optimal Bitcoin price prediction model among the three proposed models, namely LSTM, GRU and Prophet along with the implementation of various architectural configurations and hyperparameter tuning.

1.1 Introduction

Bitcoin is the world's first decentralized digital crypto currency that one can buy, sell and exchange directly without the intervention of central authorities such as banks, or government. Satoshi Nakamoto, a group of anonymous developers introduced cryptocurrency in 2009. Fiat currencies are issued and regulated by governments or central authorities. Cryptocurrencies, on the other hand, are decentralized, and powered by a technology called blockchain. It is not owned by anyone. All bitcoin transactions are public, traceable, and immutable. When a user sends a payment, the transaction - a digitally signed message using cryptography is sent to all nodes in the Bitcoin network for verification. A group of participants know as miners, verify, and timestamp the transactions into a shared and distributed database, the blockchain. The miners are then rewarded freshly mined bitcoins for contributing their computing resources in validating the transactions. Bitcoin can be obtained by mining or trading, for example exchanging for products, services, or other currencies.

1.2 Motivation

Nowadays, cryptocurrencies are becoming a global phenomenon that attracts a significant number of investors worldwide. There are more than 6,000 different types of cryptocurrencies. Being the most popular, most valuable, and highest market capitalization, combined with its decentralized and immutable properties, Bitcoin shows a promising future. As of August 2022, the number of Bitcoin wallet users on Blockchain.com have reached 84 million users. That is approximately 43 million users more than in 2019 (a 93% increase). According to BUY BITCOIN WORLDWIDE, 65%

of cryptocurrency users are Bitcoin owners. Bitcoin has received a lot of attention due to its steep increased in value, from 2020 quarter 3 to 2021 quarter 1. Its value in November 2021 exceeded over 65,000 USD (approximately 291,000 MYR) and gradually falling as shown in figure 1.2.

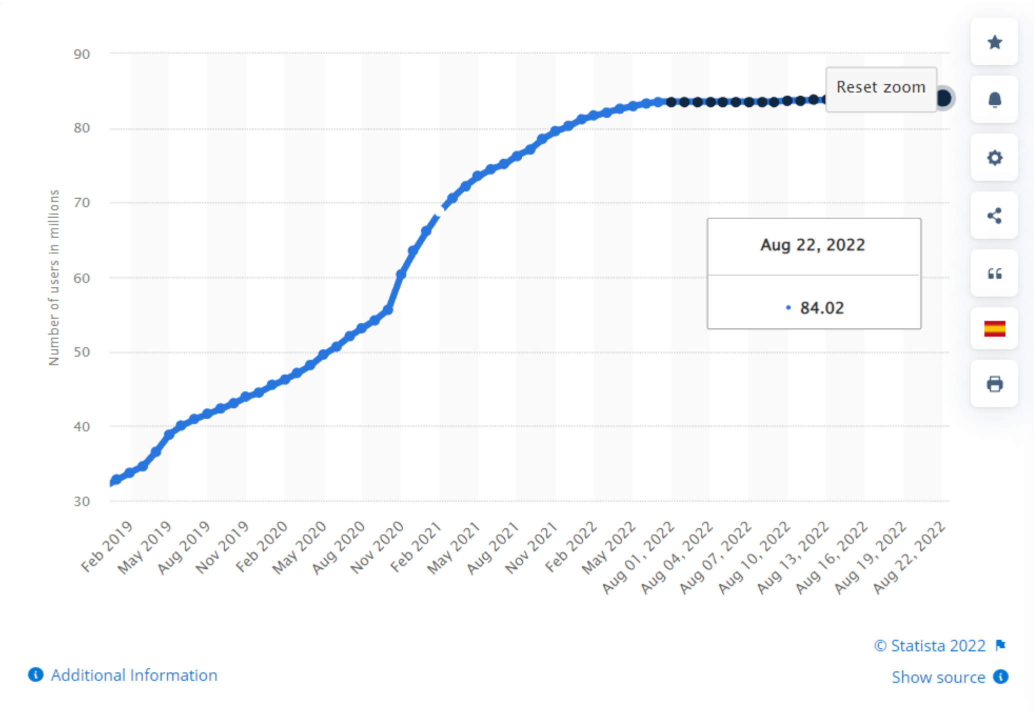


Figure 1.1: Number of Bitcoin wallet users on Blockchain.com

Figure 1.2: Bitcoin Price (Google Finance)

1.3 Problem Statement

The popularity of Bitcoin is growing rapidly, and Bitcoin is becoming a major money maker and an alternative to stock market because of its high availability and easy investment platforms. The community is using bitcoin for investment and trading. Investors are highly interested in knowing the future situation of Bitcoin to make profit. Despite being the major crypto currency, it is very risky to invest in Bitcoin because of Bitcoin's highly volatile nature and fluctuations. (Fluctuation refers to a situation in which prices go up and down.) This caught the attention of many machine learning researchers and data scientists. Past studies have used traditional time series algorithms and deep learning algorithms such as ARIMA, LSTM and GRU to forecast Bitcoin price.

1.4 Project Scope

The thesis aims to investigate and develop a researched-based Bitcoin closing price prediction model using LSTM, GRU, and Facebook Prophet along with the implementation of various architectural configurations and evaluation methods. The scope includes data collection, data preprocessing, building model, hyperparameter tuning, evaluation, comparison and visualization of model predictions and actual Bitcoin price. Past closing price of Bitcoin is used as input for training the Bitcoin prediction models. External factors such as politics, economy, and news will not be considered.

1.5 Research Objectives

The research objectives involve evaluating and assessing the performance of LSTM, GRU and Prophet models of various architectural configurations and evaluation methods in Bitcoin price forecasting using evaluation metrics like RMSE, MAE, and MAPE. Additionally, the research aims to optimize the hyperparameters of each algorithm to improve prediction accuracy. The goal is to determine the optimal models among the three proposed algorithms.

1.6 Significance, Contribution and Novelty

Past studies mostly used traditional time forecasting and deep learning algorithms. In contrast, this study introduces an additional algorithm, Prophet and conducts a comparative analysis with LSTM and GRU. Furthermore, this study employs two different architectures of LSTM and GRU (Sequence-to-Sequence, and Sequence-to-One) and implements a sliding window approach for making predictions with Prophet. This provides insights into the strengths and weaknesses of each approach. Researchers and developers will have a broader range of options to choose for predicting Bitcoin price.

CHAPTER 2

Literature Review

Various machine learning models can be used to predict the price of Bitcoin. Time series models are widely used in time-based predictions based on historical data such as stock price prediction, weather forecasting, earthquake prediction and pattern recognition. The most used time series models are AR, MA, and ARIMA. Deep learning algorithms like RNN, GRU and LSTM has also received a lot of attention in recent years.

Amin Azari [1] applied ARIMA in forecasting Bitcoin's future values based on a 3 year long past dataset. His study reveals that the model is efficient for short term prediction, for example, 1 day. However, the model has large prediction errors for long term prediction. The study shows ARIMA is unable to predict sharp fluctuations. I. M Wirawan et al. [2] study also build an ARIMA Bitcoin price prediction model based on data from 2013 to 2019. They perform hyperparameter tuning on the model and found out ARIMA (4,1,4) has the performance with the smallest MAPE value, 0.87 for the first day prediction and 5.98 for the seventh day prediction. They stated ARIMA model is better for short term predictions.

In Shaily Roy et al. - "Bitcoin Price Forecasting Using Time Series Analysis" [3]. These researchers focused on creating a consistent time series to predict Bitcoin's closing price for the next 10 consecutive days by applying different time series analysis models, namely ARIMA, AR, and MA. They compared the accuracy and found that ARIMA gives the most accurate predictions. MA model has the lowest accuracy among all 3 models. However, their research is not based on live data but 4 years of bitcoin closing price dataset from July 2013 to August 2017 from coindesk.

Anshul et al. [4] applied Long Short-Term Memory (LSTM) to develop a predictive model to forecast Bitcoin price and compare the performance with existing ARIMA model of Irfan Ahmed Mohammed Saleem and Dr. S. Jaisankar (2018). Their study only focused on the closing price of Bitcoin dataset from Bitfinex Exchange from April 28, 2013, to February 2018. The data was normalized using MinMaxScaler package and then split into 67% for training set and 33% for testing set. In their study, LSTM model takes much longer time (61 millisecond) to compile because of its complex calculations than ARIMA model (4 millisecond). The loss for LSTM model is also minimal, at the learning of 0.01. They stated the lower loss of LSTM combined with LSTM's capability of recognizing longer-term dependencies make it a more suitable model for forecasting time series data of high fluctuations than ARIMA. They added, further study involving other machine learning models would confirm the result. Unfortunately, external factors such as economic, politics and news about Bitcoin were not considered and their prediction model only focused on the closing price of Bitcoin and is not based on live data but past data. Using live data would improve the performance and predictability of the model.

Model	Compilation Time (ms)
LSTM	61
ARIMA	4

Table 2.1 LSTM and ARIMA Compilation Time Comparison

McNally et al. [5] proposed 3 prediction models on the return of investment using RNNs and LSTM and ARIMA models. The dataset used were from 19th, August 2013 until 19th, July 2016 and split into 80% for training and 20% for test sets. Data standardization was chosen over normalization. RMSE is used to evaluate the accuracy of the model while precision represents how many positively classified predictions are relevant. Having precision of 100% does not mean good overall performance, but decent at identifying price direction change. Based on their study results, both RNN and LSTM models outperform ARIMA model. LSTM achieved the highest accuracy (52.78%) while RNN achieved the lowest RMSE (5.45%). ARIMA performs poorly in both accuracy and RMSE. The CPU used in their study was an Intel Core i7 2.6GHz and GPU used was an NVIDIA GeForce 940M 2GB. It was run on Ubuntu 14.04 LTS on an SSD. The models trained on GPU was faster than the CPU as shown in Figure 2.1. Their study also found that LSTM takes longer to train model than RNN due to complex calculations and computations of LSTM. The limitation of their study is that the model is not implemented in real time settings, but past dataset.

Model	Epochs	CPU	GPU
RNN	50	56.71s	33.15s
LSTM	50	59.71s	38.85s
RNN	500	462.31s	258.1s
LSTM	500	1505s	888.34s
RNN	1000	918.03s	613.21s
LSTM	1000	3001.69s	1746.87s

Figure 2.1 Performance Comparison Between RNN and LSTM models (McNally)

Temesgen et al. [6] proposed 2 predictive algorithms using LSTM and GRU (Gated recurrent unit). The dataset was collected from Kaggle, from January 1, 2014, to February 20, 2018, normalized and split into 80% for training and 20% for testing set. Based on their study, LSTM (53ms) has higher compilation time than GRU (5ms). Their result shows that GRU has lesser MSE than LSTM. Based on their study, the RMSE and MAPE of GRU for 1 day and 3 days ahead is lower than that of LSTM. However, for 5, 7, 15 days ahead, LSTM has lower RMSE and MAPE. But overall, GRU model has better overall accuracy than LSTM model. Unfortunately, external factors such as economic, politics and news about Bitcoin were not considered in their paper.

Model	**Compilation time (ms)**
LSTM	53
GRU	5

Table 2.2 LSTM and GRU Compilation Time Comparison

Window size	Number of days ahead	LSTM		GRU	
		RMSE	MAPE	RMSE	MAPE
1	1	0.092	0.068	0.075	0.065
5	3	0.079	0.057	0.065	0.046
7	5	0.081	0.060	0.087	0.062
12	7	0.045	0.030	0.051	0.035
15	15	0.067	0.048	0.067	0.058

Figure 2.2: Comparison of RMSE and MAPE values obtained using LSTM and GRU (Temesgen et al.)

Karunya et al. [7] used a live dataset with open, high, low, and closing price of Bitcoin from quandl.com and applied decision tree and linear regression model to forecast the price of Bitcoin for the next consecutive 5 days. Their experimental result shows that linear regression model outperforms decision tree by high accuracy. In their study, the dataset is collected live and stored as .csv file which is split into 80% for training set and 20% for testing set. However, their prediction is too short, only 5 days.

Isil et al. [8] builds Prophet and ARIMA Bitcoin forecasting models in R based on past data from May 2016 until March 2018. 90 days is used as test set. Their result shows that Prophet outperforms ARIMA model which has lower RMSE, RMPSE, MAPE, MSE, and MAE values.

The past studies suggest that deep learning algorithms like LSTM, GRU and RNN indeed outperform the traditional machine learning algorithm like ARIMA, MA and AR. There is not much research about the Prophet performance. Therefore, this research paper aims to compare the Bitcoin price prediction performance of deep learning algorithms LSTM and GRU with Prophet.

CHAPTER 3

Proposed Method/Approach

This section provides an overview of the devices utilized and the algorithms employed for training the models.

3.1 System Requirement

3.1.1 Device

The device utilized in this research project is a laptop with the following specifications.

Description	Specifications
Model	Acer Nitro 5
Processor	AMD Ryzen 7 3750h (4 cores 2.5GHz base)
Operating System	Windows 10 Home
Graphics Processing Unit	Nvidia GTX 1650m (4GB Vram)
Memory	12GB
Storage	500GB SDD

Table 3.1 Device Specifications

3.2 Machine Learning Algorithms

3.2.1 LSTM

Long short-term memory (LSTM) [9] is well known for time series forecasting. LSTM was introduced by Hochreiter and Schmidhuber (1997) and gained a lot of popularity in the past few years. It is designed to overcome long term dependency problem and is capable of remembering the historical patterns for long periods.

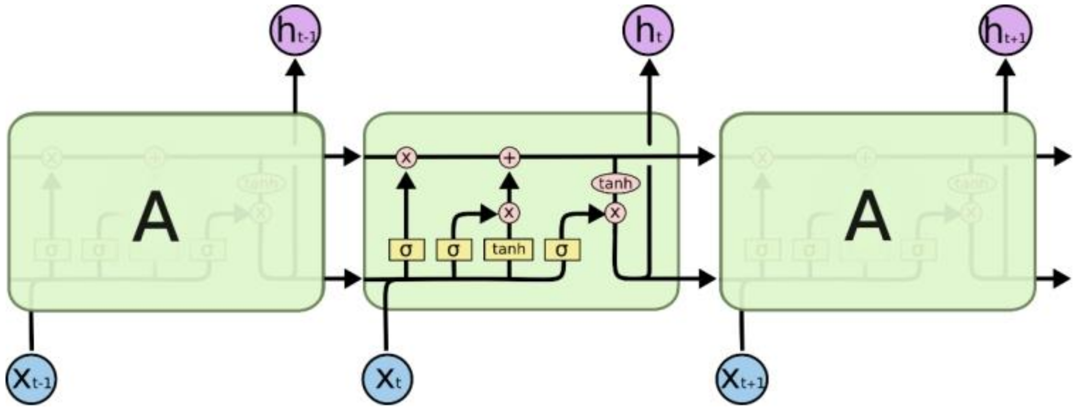
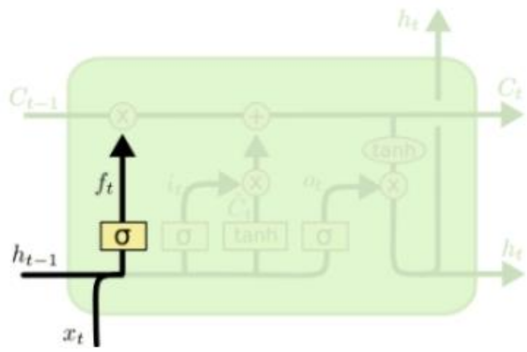


Figure 3.1: LSTM cell

LSTM contains 3 gates: forget gate, input gate and output gate. Below shows how LSTM works:

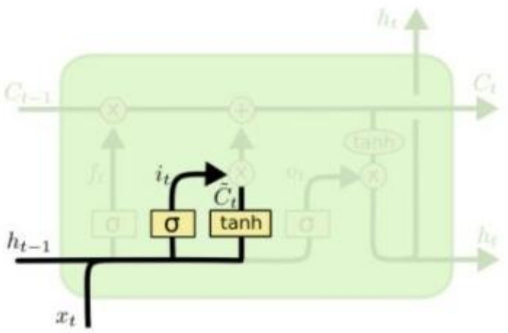
Step 1: Forget gate – decides which information to throw away from the current cell state. The information from the current input (xt) and previous hidden state (ht-1) is passed through a sigmoid function where 1 means pass through, and 0 means forget.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Figure 3.2: LSTM Forget Gate

Step 2: Input gate – decides which new information to add and store in the cell state. Them, the previous hidden state and current input are passed into the sigmoid function to decide which value will be updated, by transforming the values to [0,1]. 0 means not important while 1 means important. The same inputs are passed into tanh and squashed value to [-1, 1]. The sigmoid output and tanh output are multiplied.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 3.3: LSTM Input Gate

Step3: Cell state is multiplied by forget gate output. Possible dropping value in the cell state if forget gate approaches 0. Take the output of input gate and update the cell state to new values. New cell state is formed.

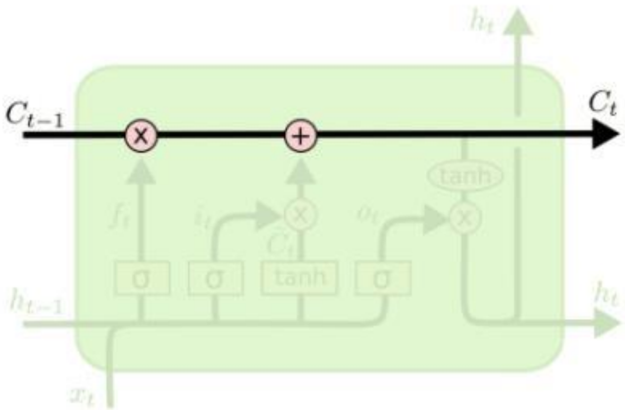
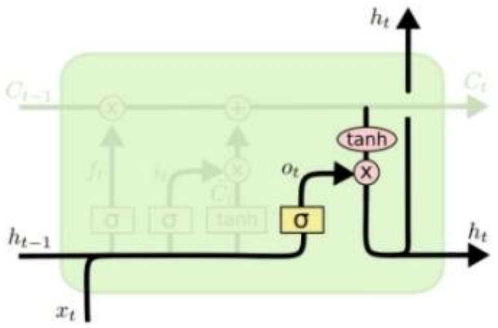


Figure 3.4: LSTM Cell State

Step 4: Output gate – determines the value of the next hidden state. Current input and previous hidden state are passed into the sigmoid function. The newly formed cell state is passed into the tanh function. Finally, the tanh output multiplies the sigmoid output and the product is the next hidden state.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 3.5: LSTM Output Gate

3.2.2 GRU

Gated recurrent unit (GRU) [10] is introduced by Kyunghnyun Cho in 2014. It is similar to LSTM with only 2 gates, forget gate and reset gate. Unlike LSTM, GRU does not have a cell state due to its simpler architecture.

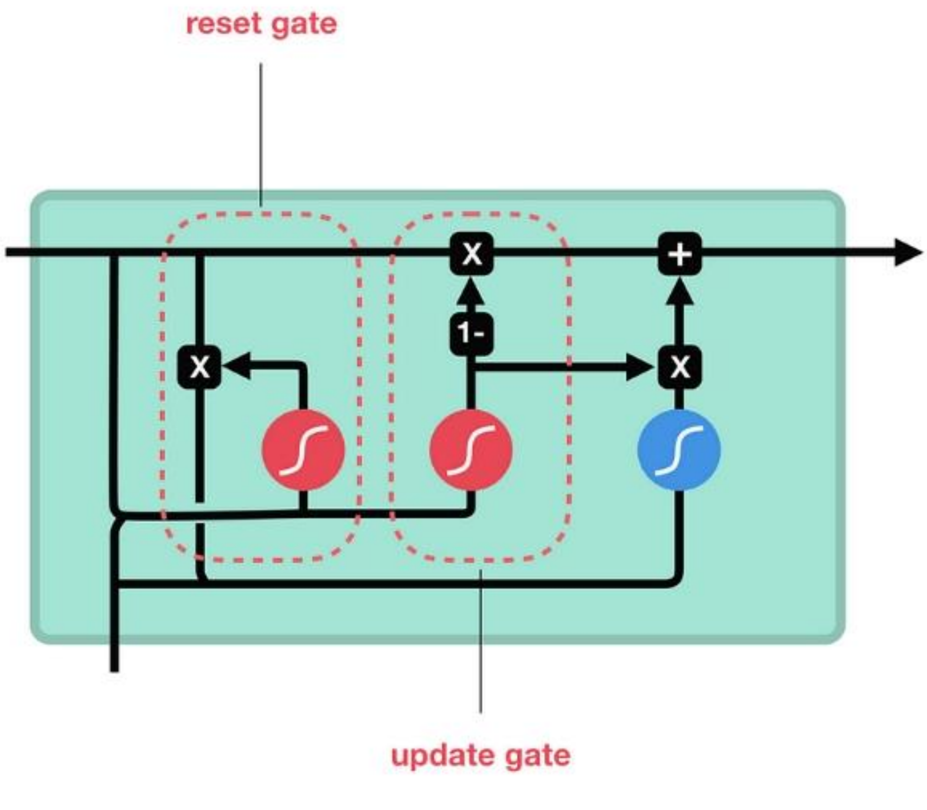


Figure 3.6 GRU cell

The reset gate is responsible for deciding what past information to forget from the previous hidden state. The purpose of update gate is to decide how much of new information to be passed.

3.2.3 Facebook Prophet

Facebook prophet [11] is an open-source algorithm for time series forecasting developed by Facebook data science team and released in 2017. Prophet is based on decomposable addictive model. The data frame needs two column, ds and y. which stores the date time series and the corresponding time series values respectively. The equation is:

$$y(t) = g(t) + s(t) + h(t) + e(t)$$

where,

- g(t) refers to the trend
- s(t) refers to the seasonality
- h(t) refers to the effects of holidays
- e(t) refers to the unconditional changes

y(t) is the forecast

CHAPTER 4

Preliminary Work

This section outlines the preliminary work and accomplishments.

4.1 Setting Up

The construction of Bitcoin price prediction models involves the utilization of Python programming language.

4.1.1 Software

Prior to building the Bitcoin price prediction models, it is essential to have the following software installed and account registered:

1. Anaconda 3
2. Jupyter Notebook
3. Google Colab account

4.1.2 Libraries

The following python libraries are essential:

- math
- pandas
- numpy
- matplotlib
- tensorflow
- prophet
- yfinance
- sklearn
- os
- google.colab

4.2 Methodology

4.2.1 Data Collection

To begin, 365 days of Bitcoin dataset from 2022-04-02 until 2023-04-01 with 1 day interval is imported from Yahoo yfinance to Jupyter Notebook.

4.2.2 Data Preprocessing

Based on the collected dataset, it consists of several columns such as Open, High, Low, Close, Adjacent Close, and Volume. Only the close price of Bitcoin is considered in this research paper. For LSTM and GRU, the remaining column in the dataset is excluded. For Prophet, the date index is converted to column, the Date and Close columns are retained and renamed to '*ds*' and '*y*' respectively. The subsequent step is to identify the number of missing values in the dataset and there is no missing value.

4.2.3 Data Splitting

In the context of machine learning, x_train and y_train refers to the input features and output values used for training models. The dataset is split into 2 subsets - training and test set. Training set refers to the subset of a dataset used to fit (train) a model while test set refers to the subset to assess the performance of the trained model. In simple words, training set teaches the learning algorithm how to make prediction or perform specific task and test set tests the trained model. The dataset is split into 335 days for training data and 30 days for testing data. The training set contains data from 2, April 2022 to 2, March 2022 while the testing set contains data from 3, March, 2023 to 1, April 2023.

4.2.4 Feature Scaling

LSTM and GRU	**Prophet**
Standard scaler is used to standardize the training data	No feature scaling is performed

Table 4.1 Data Normalization

The equation of standard scaler is as follow:

$$z = \frac{x - \mu}{s}$$

where,

μ = mean

s = standard deviation

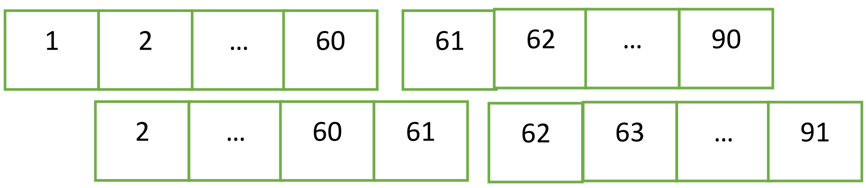
4.2.5 Time Series Data

The architectures of LSTM, GRU and Prophet are different. LSTM and GRU requires both x_train and y_train for model training. Each sample of x_train is a sequence of historical data points, whereas y_train contains the next value for each sequence in x_train. LSTM and GRU are designed to learn patterns and relationships within these sequences.

In contrast, Prophet does not require a seperate y_train but the x_train contains two columns – namely ‘ds’ the timestamp and ‘y’, the target variables. Prophet utilizes the ‘y’ values to train model and learn the patterns and seasonality in the time series data.

Sequence-to-One time series data:

The dataset is divided into input windows and outputs. The training data length is 335. In this case, *m* = 60 days for input sequence, x_train and 1 day for output, y_train. This means that the previous 60 days of data will be used to predict the value for the next day. This adopts a sliding window approach, that is from (1 until m) for x_train, and (m + 1) for y_train and the window moves forward one step at a time, until (335 – m – 1 until 334) for x_train, and (335) for y_train. A total of 275 chunks of data are used to train the models. These chunks of data are reshaped into a 3-Dimentional array with 1 feature. The shapes below illustrate the sliding window.



Sequence-to-Sequence time series data:

The dataset is divided into input windows and outputs. The training data length is 335. In this case, *m* = 60 days for input sequence, x_train and 30 days for output, y_train. This means that the previous 60 days of data will be used to predict the value for the next consecutive 30 days. This adopts a sliding window approach, that is from (1 until m) for x_train, and (m + 1 until m + 30) for y_train and the window moves forward one step at a time, until (335 – m – 1 – 30 until 304) for x_train, and (305 until 335) for y_train. A total of 245 chunks of data are used to train the models. These chunks of data are reshaped into a 3-Dimentional array with 1 feature. The shapes below illustrate the sliding window.

1	2	…	60		61
	2	…	60	61	62

Prophet time series data

The first 335 days of data was used for training. The date column is renamed to ds and closing price column is renamed to y. The rest of the columns are excluded.

4.2.6 Evaluation Metrics

The models are evaluated using RMSE, MAPE, and MAE.

- Root mean squared error, abbreviated RMSE, is the root of the mean of the difference between actual and predicted values.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}$$

Where

Y_i = actual values

$\hat{Y_i}$ = predicted values

N = the number of samples

- Mean absolute percentage error (MAPE) is the mean of absolute percentage errors of a prediction. The lower the better.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N} | \frac{Y_i - \hat{Y_i}}{Y_i} |$$

Where

Y_i = actual values

$\hat{Y_i}$ = predicted values

N = the number of samples

- Mean absolute error (MAE) is the mean of absolute errors between actual and predicted values. The lower the better.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} | \frac{Y_i - \hat{Y_i}}{N} |$$

Where

Y_i = actual values

$\hat{Y_i}$ = predicted values

N = the number of samples

4.2.7 Model

The three models used for time series forecasting of Bitcoin price in this research are:

- LSTM
- GRU
- Prophet

The default settings were used to train the models and grid search hyperparameter tuning was performed. The following hyperparameter will be used in grid search:

LSTM and GRU hyperparameter:

1. Layer 1 unit represents the number of cells in the first layer that process the input.
2. Layer 2 unit refers to the number of cells in the second layer that process the output of the first layer.
3. Epochs refers to the number of complete passes through of the entire training dataset into the model. For example, 2 epochs mean the model learn the dataset 2 times.

Prophet:

1. Changepoint prior scale controls the flexibility of the model in detecting changepoints in time series data.
2. Seasonality prior scale controls the strength of the seasonal patterns captured.
3. Holidays prior scale influences the impact of holidays on the model prediction.

CHAPTER 5

Architectural Configurations and Experiments

Hyperparameters tuning was conducted using grid search and the models were evaluated against the actual closing price using the evaluation metrics RMSE. MAPE, and MAE.

Two architectural configurations:

1. Sequence-to-Sequence where a sequence of input data is utilized to predict a sequence of output data.
2. Sequence-to-One where a sequence of input data is utilized to predict an output data.

Two evaluation methods:

1. Walk forward where the predicted output data is appended to the input batch and removing the initial data points while making it consistent length and move a step forward to predict the next values.
2. Rolling origin uses the sliding window approach and the historical data for training.

Three experiments were conducted in this research:

1. Sequence-to-Sequence Walk Forward
2. Sequence-to-One Walk Forward
3. Sequence-to-One Rolling Origin

5.1 Experiment 1: Sequence-to-Sequence Walk Forward

For LSTM and GRU, this approach employs a sequence-to-sequence time series data for model training (245 chunks of data). 25 distinct model configurations were explored for each model. Layer 1 and layer 2 units with the range of values (16, 32, 50, 64, and 128). The dense layer was set to 30. In addition, a custom early stopping mechanism was implemented to monitor the training process and halt the training if the loss does not improve for five consecutive epochs. This helps prevent overfitting and saves time by stopping the training process when further optimization is unlikely to significantly improve the model's performance. The number of epochs was set to 200. The last 60 days of training data was used as input to forecast the next consecutive 30 days of Bitcoin's closing price.

The default architecture configuration of Prophet is Sequence-to-Sequence, the first 335 days of data was used as training data to train a model and predict the next consecutive 30 days of Bitcoin's closing price. 245 distinct model configurations were explored. The hyperparameters are as follow:

- Changepoint prior scale of range of values (0.005, 0.01, 0.05, 0.1, 0.5)
- Seasonality prior scale of range of (0.01, 0.05, 0.1, 0.5, 1, 5, 10)
- Holidays prior scale of range of (0.01, 0.05, 0.1, 0.5, 1, 5, 10)

5.1.1 Experiment 1: LSTM

Sorted RMSE

	layer 1 units	layer 2 units	epochs	RMSE	MAPE	MAE
0	16	16	72	3730.464174	12.204144	3182.360725
16	64	32	35	4858.761910	13.802989	3685.919332
15	64	16	23	4942.487709	14.058138	3754.107609
23	128	64	20	5184.401413	15.133758	4051.472461
8	32	64	26	5247.463257	15.254666	4089.222001

Sorted MAPE

	layer 1 units	layer 2 units	epochs	RMSE	MAPE	MAE
0	16	16	72	3730.464174	12.204144	3182.360725
16	64	32	35	4858.761910	13.802989	3685.919332
15	64	16	23	4942.487709	14.058138	3754.107609
23	128	64	20	5184.401413	15.133758	4051.472461
18	64	64	16	5275.956847	15.195657	4076.096003

Sorted MAE

	layer 1 units	layer 2 units	epochs	RMSE	MAPE	MAE
0	16	16	72	3730.464174	12.204144	3182.360725
16	64	32	35	4858.761910	13.802989	3685.919332
15	64	16	23	4942.487709	14.058138	3754.107609
23	128	64	20	5184.401413	15.133758	4051.472461
18	64	64	16	5275.956847	15.195657	4076.096003

Figure 5.1.1.1 LSTM Sequence-to-Sequence Sorted RMSE, MAPE, and MAE

The dataframe from the figure above shows that LSTM Sequence-to-Sequence model with 16 layer 1 unit, 16 layer 2 unit and 72 epochs is the optimal model with the lowest RMSE (3730.464), MAPE (12.204), and MAE (3182.361) among 25 models.

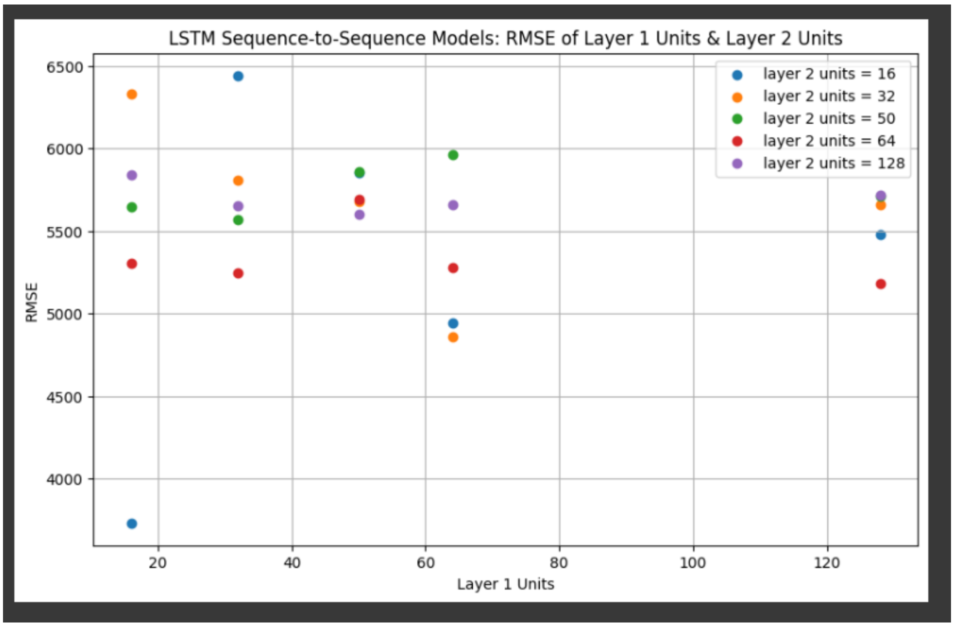


Figure 5.1.1.2 LSTM Sequence-to-Sequence Models RMSE Visualization

The figure above shows that LSTM Sequence-to-Sequence model with 16 layer 1 unit and 16 layer 2 unit has the lowest RMSE.

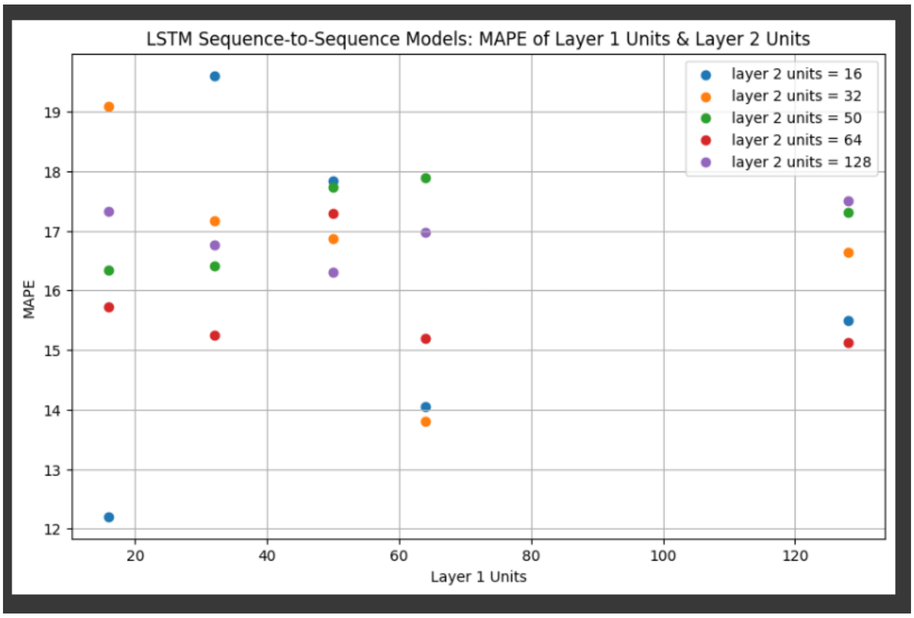


Figure 5.1.1.3 LSTM Sequence-to-Sequence Models MAPE Visualization

The figure above shows that LSTM Sequence-to-Sequence model with 16 layer 1 unit and 16 layer 2 unit has the lowest MAPE.

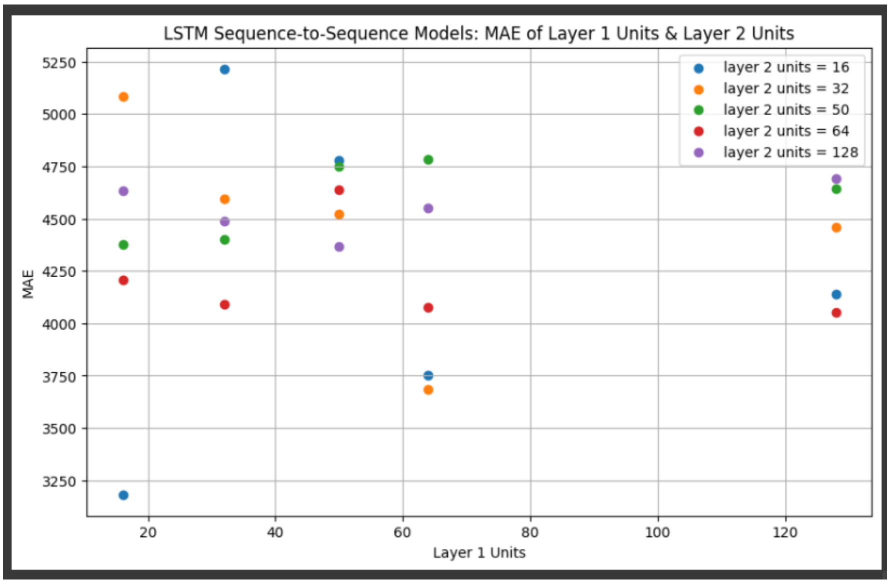


Figure 5.1.1.4 LSTM Sequence-to-Sequence Models MAE Visualization

The figure above shows that LSTM Sequence-to-Sequence model with 16 layer 1 unit and 16 layer 2 unit has the lowest MAE.

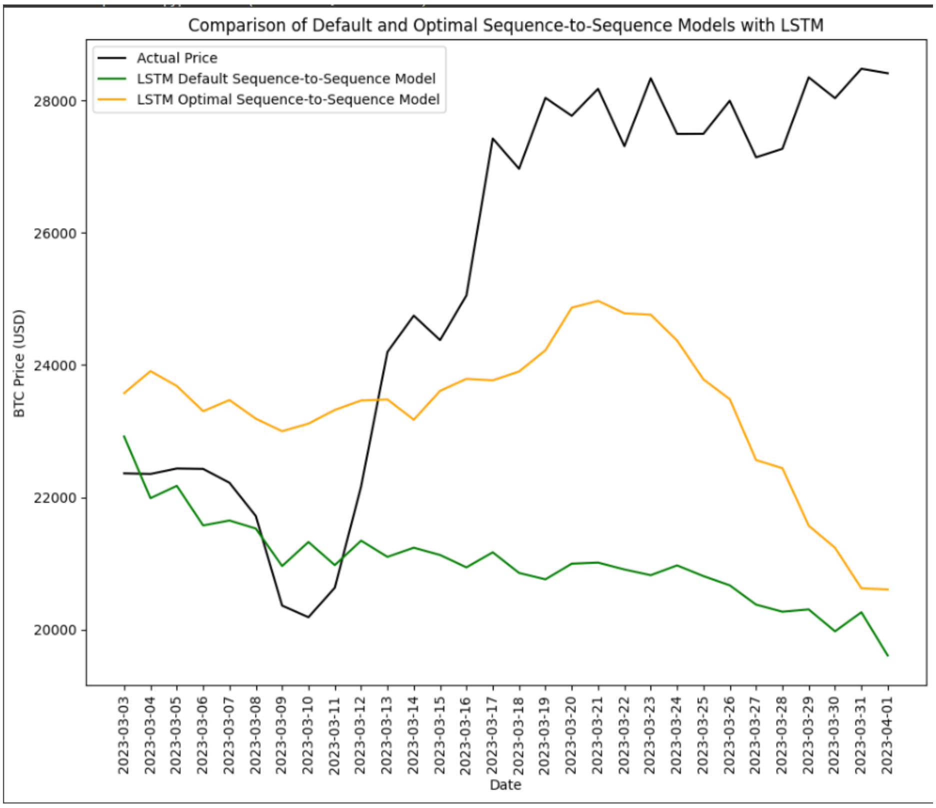


Figure 5.1.1.5 Comparison of Default and Optimal Sequence-to-Sequence Models with LSTM

	RMSE	MAPE	MAE
LSTM Default Sequence-to-Sequence Model	5439.214	16.6140	4458.792
LSTM Optimal Sequence-to-Sequence Model	3730.464	12.204	3182.361

Table 5.1.1.1 Comparison of Default and Optimal Sequence-to-Sequence Models with LSTM

Figure 5.1.1.1 shows the comparison of predicted price and actual price and table above show the comparison of Default and Optimal Sequence-to-Sequence Models with LSTM. Clearly, the optimal model outperforms the default model with lower RMSE, MAPE, and MAE.

5.1.2 Experiment 1: GRU

Sorted RMSE

	layer 1 units	layer 2 units	epochs	RMSE	MAPE	MAE
6	32	32	108	4945.610650	15.388271	4135.137370
21	128	32	53	5346.172839	15.861925	4255.525690
10	50	16	48	5366.640639	15.302280	4094.001849
14	50	128	37	5463.138367	17.389718	4660.301667
18	64	64	74	5466.157716	16.723454	4470.729540

Sorted MAPE

	layer 1 units	layer 2 units	epochs	RMSE	MAPE	MAE
10	50	16	48	5366.640639	15.302280	4094.001849
6	32	32	108	4945.610650	15.388271	4135.137370
21	128	32	53	5346.172839	15.861925	4255.525690
15	64	16	37	5525.833634	15.891342	4254.521597
20	128	16	94	5646.984571	15.976251	4256.600308

Sorted MAE

	layer 1 units	layer 2 units	epochs	RMSE	MAPE	MAE
10	50	16	48	5366.640639	15.302280	4094.001849
6	32	32	108	4945.610650	15.388271	4135.137370
15	64	16	37	5525.833634	15.891342	4254.521597
21	128	32	53	5346.172839	15.861925	4255.525690
20	128	16	94	5646.984571	15.976251	4256.600308

Figure 5.1.2.1 GRU Sequence-to-Sequence sorted RMSE, MAPE, and MAE

The dataframe from the figure above shows that GRU Sequence-to-Sequence model with 50 layer 1 unit, 16 layer 2 unit and 48 epochs is the optimal model with the third lowest RMSE (5366.641), lowest MAPE (15.302), and lowest MAE (4094.001) among 25 models.

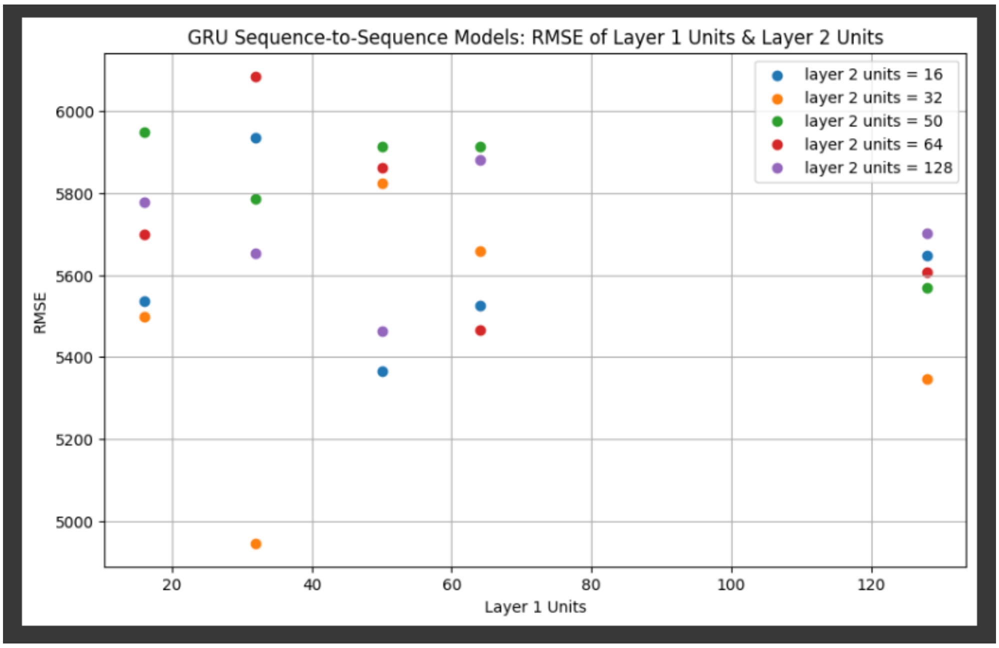


Figure 5.1.2.2 GRU Sequence-to-Sequence Models RMSE Visualization

The figure above shows that GRU Sequence-to-Sequence model with 32 layer 1 unit and 32 layer 2 unit has the lowest RMSE while the model with 50 layer 1 unit and 16 layer 2 unit has the third lowest RMSE

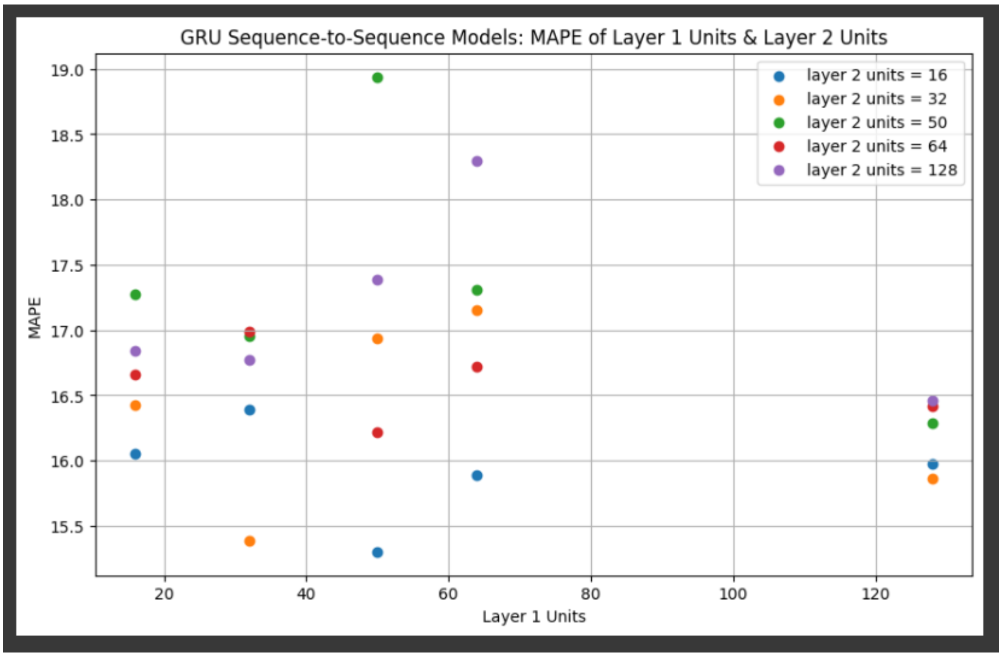


Figure 5.1.2.3 GRU Sequence-to-Sequence Models MAPE Visualization

The figure above shows that GRU Sequence-to-Sequence model with 50 layer 1 unit and 16 layer 2 unit has the lowest MAPE.

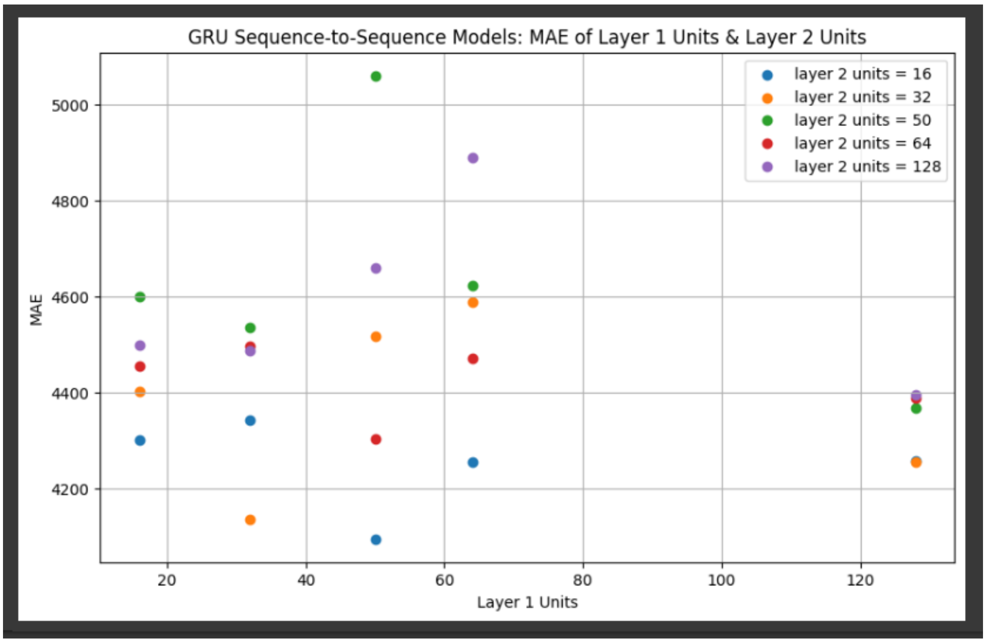


Figure 5.1.2.4 GRU Sequence-to-Sequence Models MAE Visualization

The figure above shows that GRU Sequence-to-Sequence model with 50 layer 1 unit and 16 layer 2 unit has the lowest MAE.

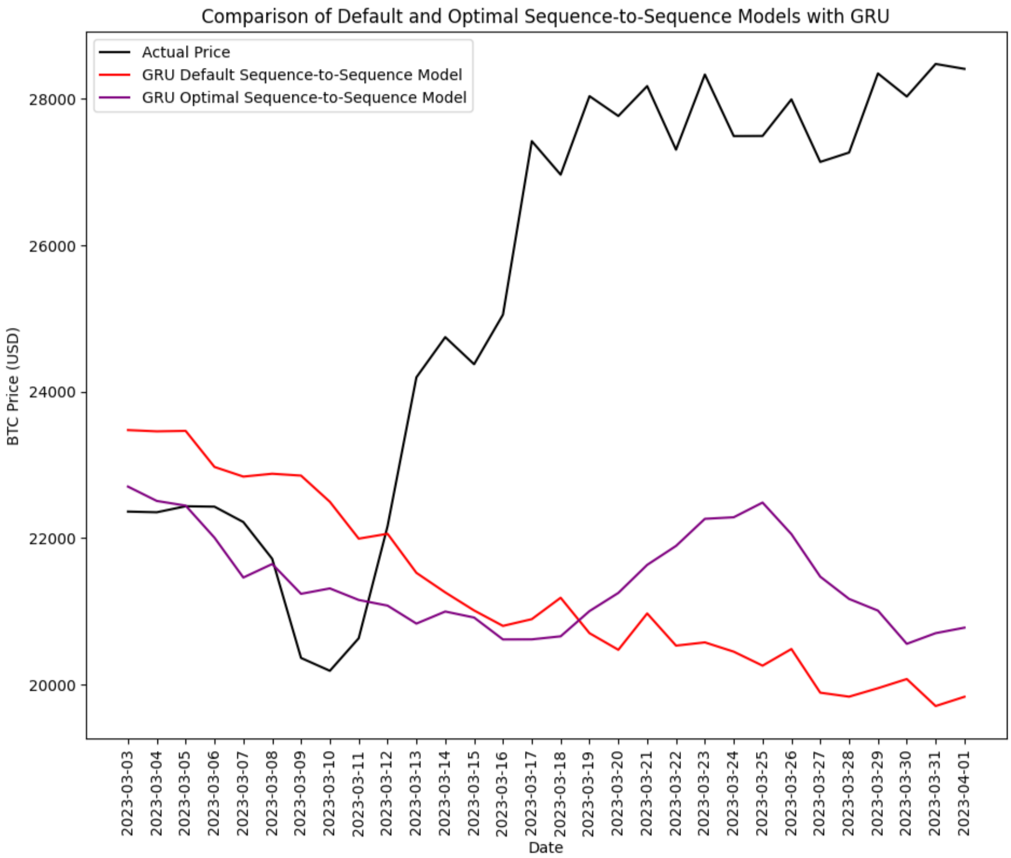


Figure 5.1.2.5 Comparison of Default and Optimal Sequence-to-Sequence Models with GRU

	RMSE	MAPE	MAE
GRU Default Sequence-to-Sequence Model	5655.136	16.245	4340.016
GRU Optimal Sequence-to-Sequence Model	4945.611	15.388	4135.137

Table 5.1.2.1 Comparison of Default and Optimal Sequence-to-Sequence Models with GRU

Figure 5.1.2.5 shows the comparison of predicted price and actual price and table above show the comparison of Default and Optimal Sequence-to-Sequence Models with GRU. Clearly, the optimal model outperforms the default model with lower RMSE, MAPE, and MAE.

5.1.3 Experiment 1: Prophet

	changepoint prior scale	holiday prior scale	seasonality prior scale	RMSE	MAPE	MAE
122	0.05	0.50	0.50	2431.102120	7.884006	1781.179761
143	0.05	10.00	0.50	2431.102120	7.884006	1781.179761
108	0.05	0.05	0.50	2431.102120	7.884006	1781.179761
136	0.05	5.00	0.50	2431.102120	7.884006	1781.179761
115	0.05	0.10	0.50	2431.102120	7.884006	1781.179761
129	0.05	1.00	0.50	2431.102120	7.884006	1781.179761
101	0.05	0.01	0.50	2431.102120	7.884006	1781.179761
105	0.05	0.05	0.01	2457.561149	7.918719	1784.931478
112	0.05	0.10	0.01	2457.561149	7.918719	1784.931478
119	0.05	0.50	0.01	2457.561149	7.918719	1784.931478

Figure 5.1.3.1 Prophet Sequence-to-One sorted RMSE, MAPE, and MAE

The dataframe from the figure above shows that Prophet Sequence-to-One model with changepoint prior scale 0.005, seasonality prior scale 0.05 and holiday prior scale (0.01, 0.05, 0.1, 0.5, 1, 5, 10) is the optimal model with the exactly same RMSE (2431.102), MAPE (7.884), and MAE (1781.180) among 245 models.

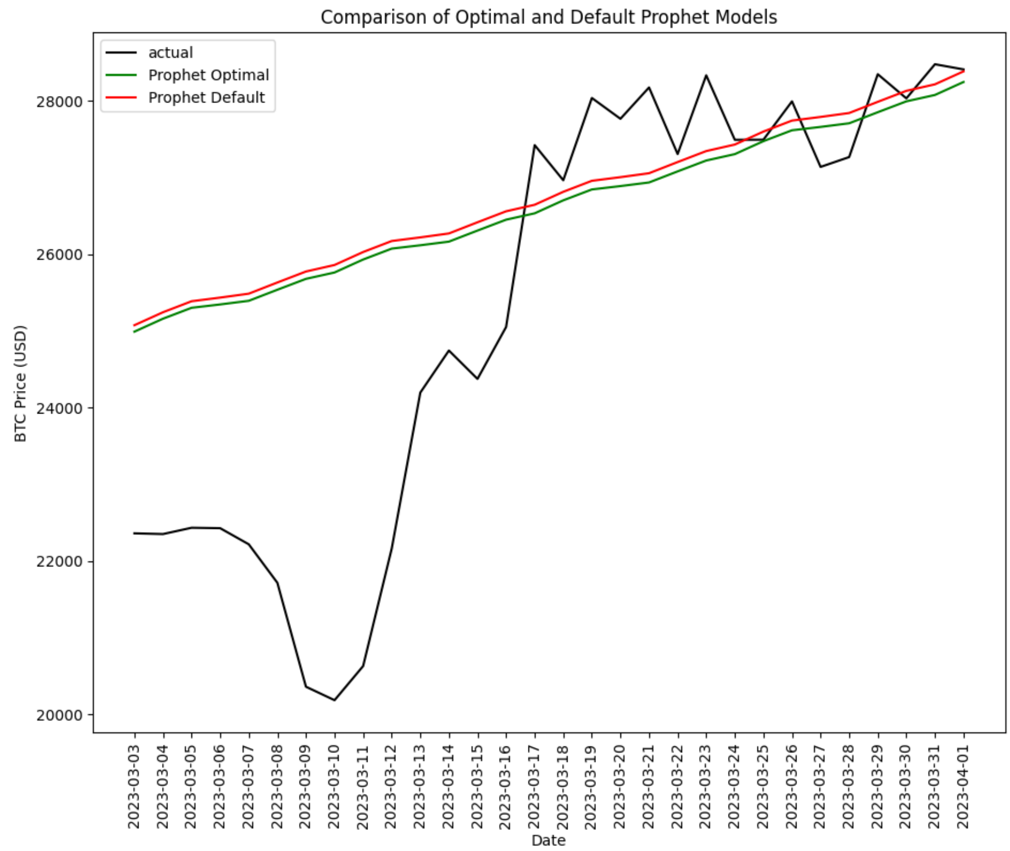


Figure 5.1.3.2 Comparison of Optimal and Default Prophet Models

	RMSE	MAPE	MAE
Prophet Default Model	2481.383	7.957	1790.358
Prophet Optimal Model	2431.102	7.884	1781.180

Table 5.1.3.1 Comparison of Optimal and Default Sequence-to-Sequence Prophet Models

The Figure 5.1.3.2 shows that the optimal model of Prophet of default architectural configurations is slightly better than the default model. However, it looks like the prophet of the default architectural configurations is not capable of predicting curvature and trend of time series data.

5.2 Experiment 2: Sequence-to-One Walk Forward

For LSTM and GRU, this approach employs a sequence-to-one time series data for model training (275 chunks of data). 25 distinct model configurations were explored for each model. Layer 1 and layer 2 units with the range of values (16, 32, 50, 64, and 128). The dense layer was set to 1. In addition, a custom early stopping mechanism was implemented to monitor the training process and halt the training if the loss does not improve for five consecutive epochs. This helps prevent overfitting and saves time by stopping the training process when further optimization is unlikely to significantly improve the model's performance. The number of epochs was set to 100. The input batch consists of the last 60 days of training data was used to forecast the next day of Bitcoin's closing price. This prediction was appended to the input batch, while excluding the first data in the batch, making it consistent 60 data in the batch. This newly created batch was used to predict the next day of Bitcoin's closing price again. The process is iteratively repeated, with the sliding window approach, moving one step forward, until it accomplished 30 days of prediction.

It is worth noting that the architectures of Prophet and the two deep learning model LSTM and GRU are different. Hence, the way of training model is also different. In this sequence-to-one approach, Prophet does not require output variables and does not train on the entire training data, but train on the last 30 days of data as the training data to predict the next day value. Similarly, this prediction was appended to the training data, while excluding the first data, making it consistent 60 training data. This newly created training data was used to predict the next day of Bitcoin's closing price again. The process is iteratively repeated, with the sliding window approach, moving one step forward, until it accomplished 30 days of prediction. 35 distinct model configurations were explored. The hyperparameters are as follow:

- Changepoint prior scale of range of values (0.005, 0.01, 0.05, 0.1, 0.5)
- Seasonality prior scale of range of (0.01, 0.05, 0.1, 0.5, 1, 5, 10)

	changepoint prior scale	holiday prior scale	RMSE	MAPE	MAE
0	0.005	0.01	2605.299672	9.731863	2435.928681
1	0.005	0.05	2605.299672	9.731863	2435.928681
2	0.005	0.10	2605.299672	9.731863	2435.928681
3	0.005	0.50	2605.299672	9.731863	2435.928681
4	0.005	1.00	2605.299672	9.731863	2435.928681
5	0.005	5.00	2605.299672	9.731863	2435.928681
6	0.005	10.00	2605.299672	9.731863	2435.928681

Figure 5.2.1 Holiday Prior Scale

Based on the figure above, holidays prior scale was not considered in this experiment since it has no effect on the performance of the model.

5.2.1 Experiment 2: LSTM

Sorted RMSE

	layer 1 units	layer 2 units	epochs	RMSE	MAPE	MAE
24	128	128	15	1934.700035	6.852531	1646.738856
11	50	32	28	2986.492548	10.477892	2702.833419
0	16	16	37	3470.207479	11.898361	3106.927872
20	128	16	18	3517.721370	12.028223	3145.088941
14	50	128	19	3791.807112	12.732424	3352.050834

Sorted MAPE

	layer 1 units	layer 2 units	epochs	RMSE	MAPE	MAE
24	128	128	15	1934.700035	6.852531	1646.738856
11	50	32	28	2986.492548	10.477892	2702.833419
0	16	16	37	3470.207479	11.898361	3106.927872
20	128	16	18	3517.721370	12.028223	3145.088941
14	50	128	19	3791.807112	12.732424	3352.050834

Sorted MAE

	layer 1 units	layer 2 units	epochs	RMSE	MAPE	MAE
24	128	128	15	1934.700035	6.852531	1646.738856
11	50	32	28	2986.492548	10.477892	2702.833419
0	16	16	37	3470.207479	11.898361	3106.927872
20	128	16	18	3517.721370	12.028223	3145.088941
14	50	128	19	3791.807112	12.732424	3352.050834

Figure 5.2.1.1 LSTM Sequence-to-One sorted RMSE, MAPE, and MAE

The dataframe from the figure above shows that LSTM Sequence-to-One model with 128 layer 1 unit, 128 layer 2 unit and 15 epochs is the optimal model with the lowest RMSE (1934.700), MAPE (6.853), and MAE (1646.739) among 25 models.

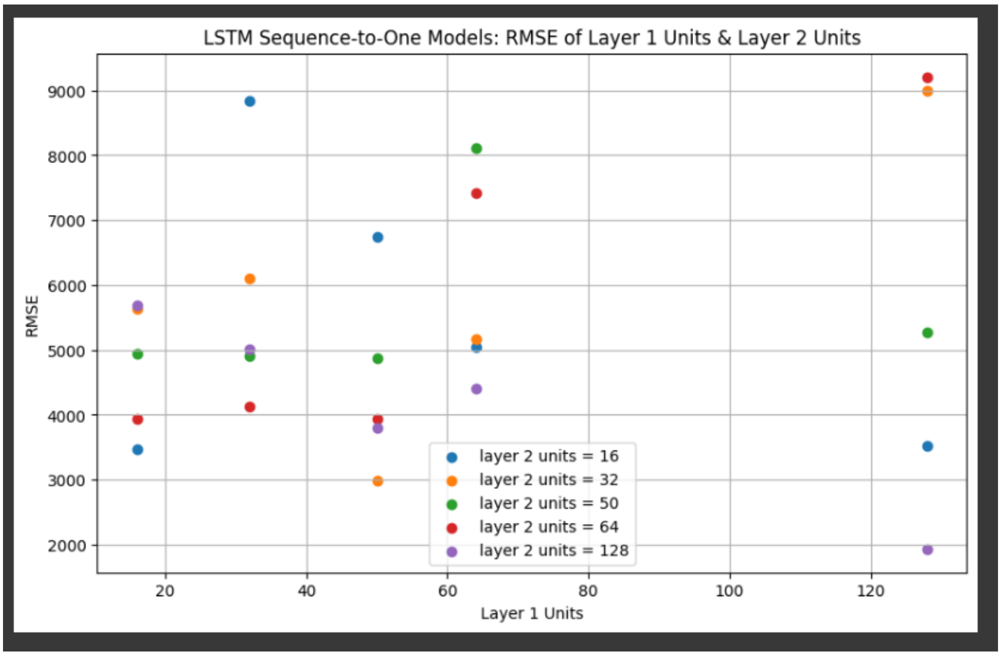


Figure 5.2.1.2 LSTM Sequence-to- One Models RMSE Visualization

The figure above shows that LSTM Sequence-to-One model with 128 layer 1 unit, and 128 layer 2 unit has the lowest RMSE.

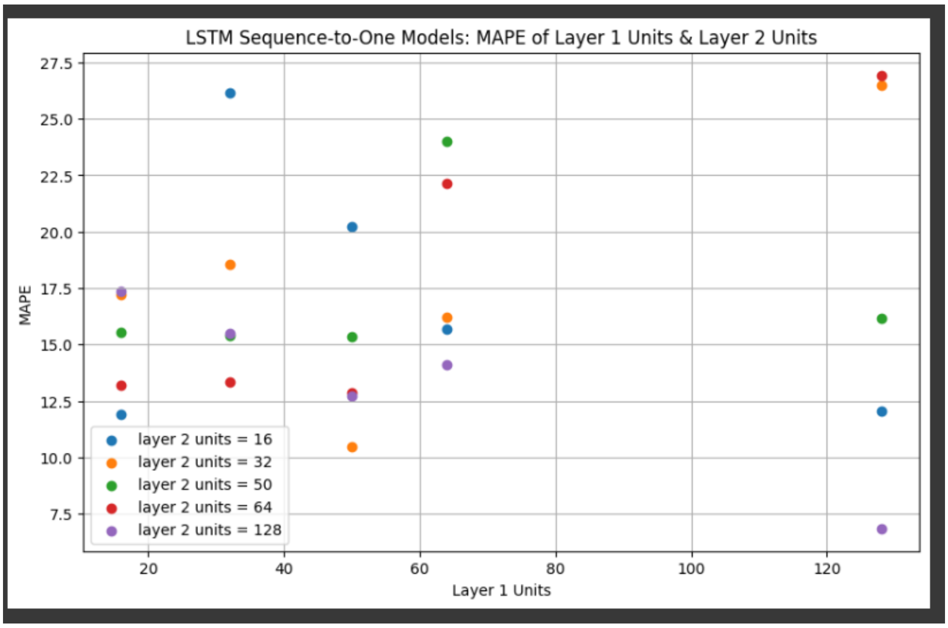


Figure 5.2.1.3 LSTM Sequence-to-One Models MAPE Visualization

The figure above shows that LSTM Sequence-to-One model with 128 layer 1 unit, and 128 layer 2 unit has the lowest MAPE.

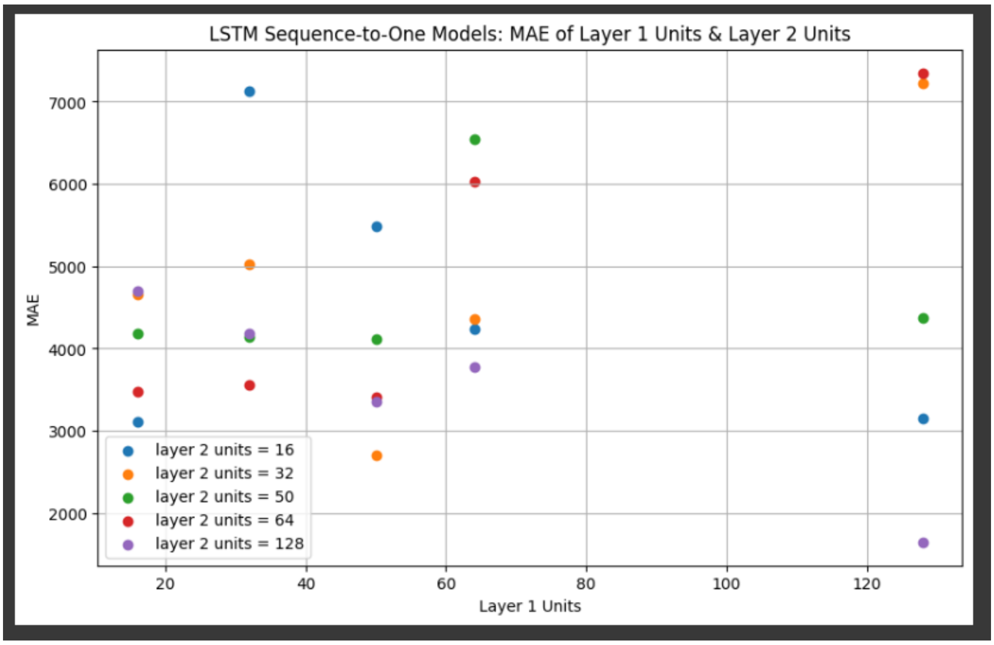


Figure 5.2.1.4 LSTM Sequence-to-One Models MAE Visualization

The figure above shows that LSTM Sequence-to-One model with 128 layer 1 unit, and 128 layer 2 unit has the lowest MAE.

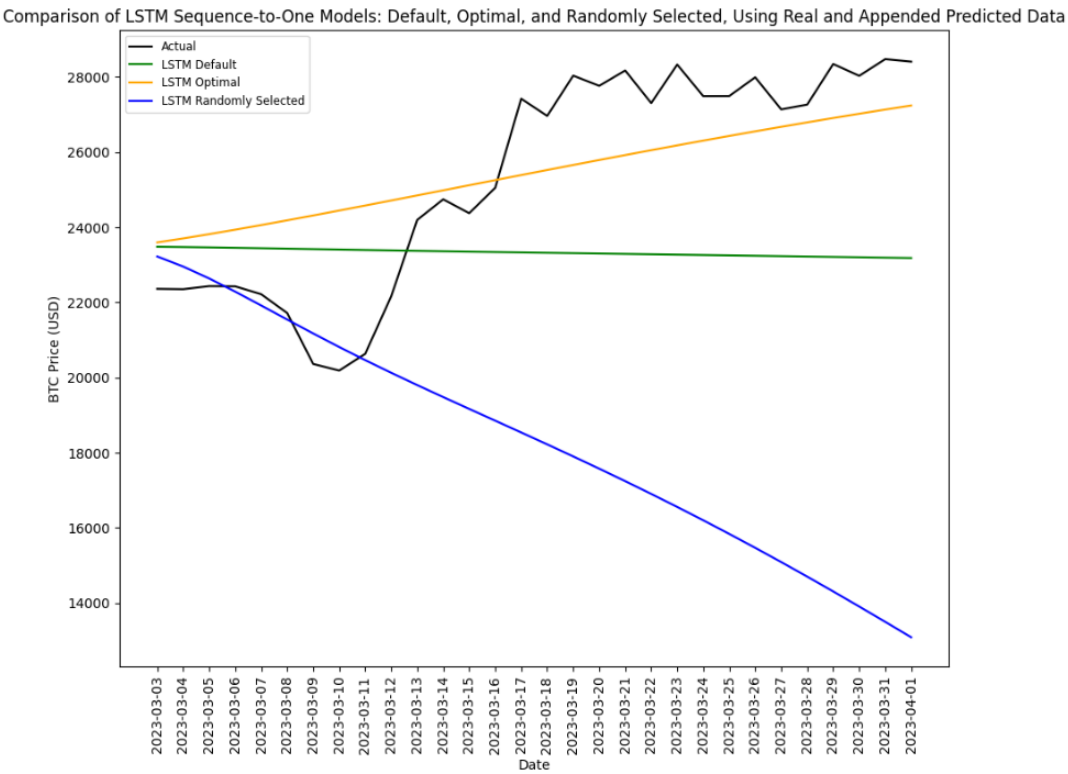


Figure 5.2.1.5 Comparison of LSTM Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Walk Forward

	RMSE	MAPE	MAE
LSTM Default Sequence-to-One Model Using Real and Appended Predicted Data	3547.077	12.103	3166.228
LSTM Optimal Sequence-to-One Model Using Real and Appended Predicted Data	1934.700	6.853	1646.739
LSTM Randomly Selected Sequence-to-One Model Using Real and Appended Predicted Data	8994.597	26.474	7221.101

Table 5.2.1.1 Comparison of LSTM Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Walk Forward

Figure 5.2.1.5 shows the comparison of LSTM Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Real and Appended Predicted Data. Based on the result in Table 5.2.1.1, the optimal model outperforms the other two with the lowest RMSE, MAPE, and MAE, the randomly selected one is the worst among the three.

5.2.2 Experiment 2: GRU

Sorted RMSE

	layer 1 units	layer 2 units	epochs	RMSE	MAPE	MAE
13	50	64	13	2396.268798	8.685752	2165.020864
22	128	50	14	3173.641253	11.033936	2858.487462
5	32	16	15	3479.401763	11.915941	3113.019507
21	128	32	9	3709.275342	12.475381	3281.865257
4	16	128	9	3815.796661	12.747248	3361.828260

Sorted MAPE

	layer 1 units	layer 2 units	epochs	RMSE	MAPE	MAE
13	50	64	13	2396.268798	8.685752	2165.020864
22	128	50	14	3173.641253	11.033936	2858.487462
5	32	16	15	3479.401763	11.915941	3113.019507
21	128	32	9	3709.275342	12.475381	3281.865257
4	16	128	9	3815.796661	12.747248	3361.828260

Sorted MAE

	layer 1 units	layer 2 units	epochs	RMSE	MAPE	MAE
13	50	64	13	2396.268798	8.685752	2165.020864
22	128	50	14	3173.641253	11.033936	2858.487462
5	32	16	15	3479.401763	11.915941	3113.019507
21	128	32	9	3709.275342	12.475381	3281.865257
4	16	128	9	3815.796661	12.747248	3361.828260

Figure 5.2.2.1 GRU Sequence-to-One Walk Forward Sorted RMSE, MAPE, and MAE

The dataframe from the figure above shows that GRU Sequence-to- One model with 50 layer 1 unit, 64 layer 2 unit and 13 epochs is the optimal model with the third lowest RMSE (2396.269), lowest MAPE (8.686), and lowest MAE (2165.021) among 25 models.

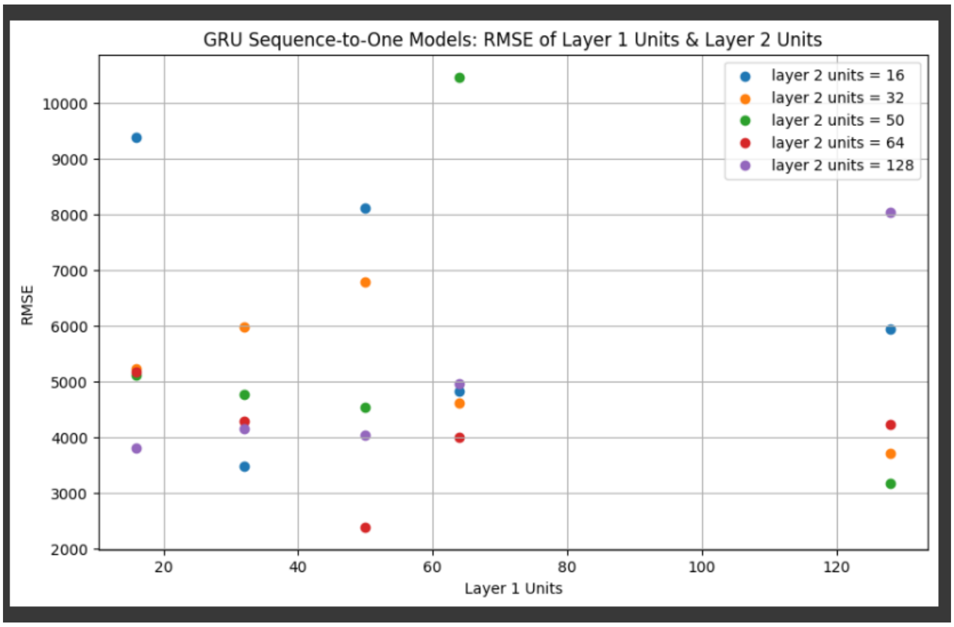


Figure 5.2.2.2 GRU Sequence-to-One Walk Forward Models RMSE Visualization

The figure above shows that GRU Sequence-to- One model with with 50 layer 1 unit, and 64 layer 2 unit has the lowest RMSE.

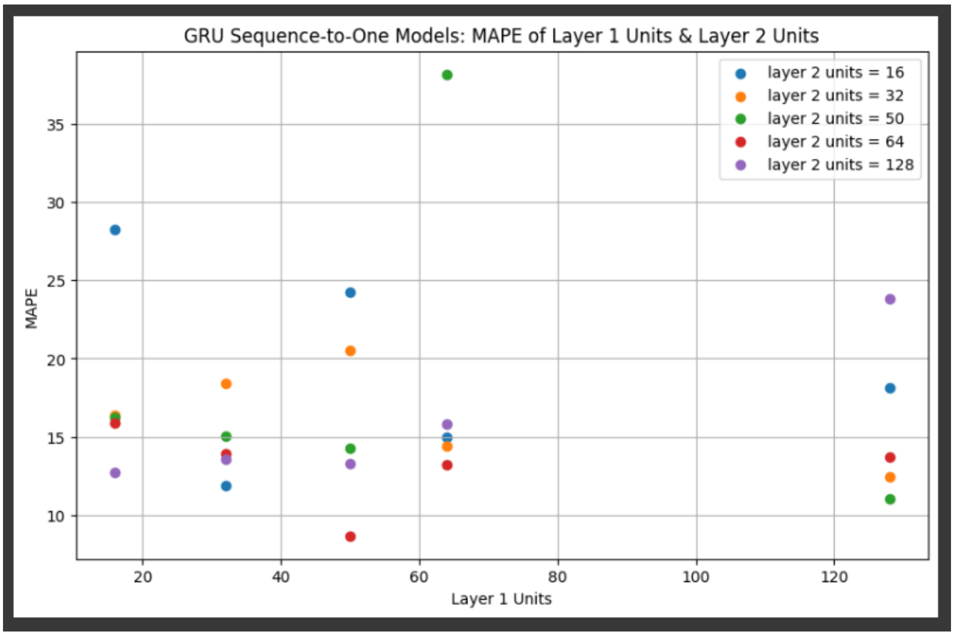


Figure 5.2.2.3 GRU Sequence-to-One Walk Forward Models MAPE Visualization

The figure above shows that GRU Sequence-to- One model with 50 layer 1 unit, and 64 layer 2 unit has the lowest MAPE.

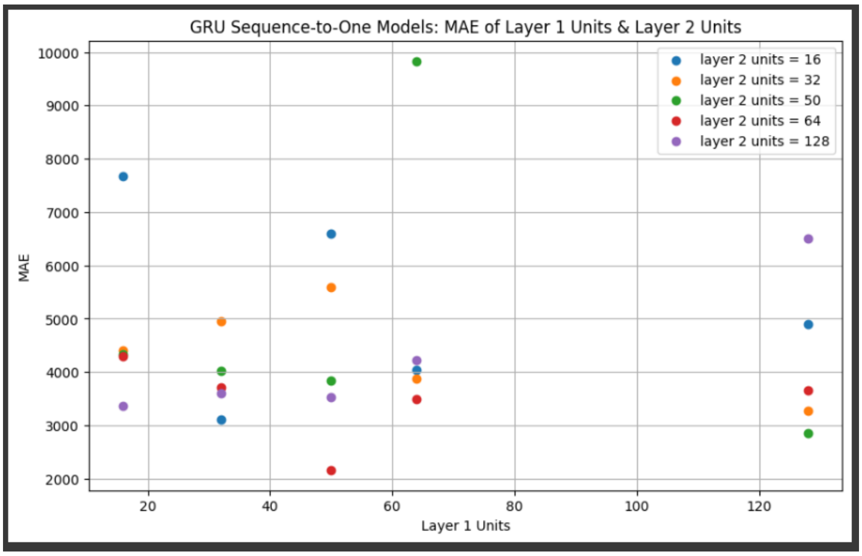


Figure 5.2.2.4 GRU Sequence-to-One Walk Forward Models MAE Visualization

The figure above shows that GRU Sequence-to- One model with 50 layer 1 unit, and 64 layer 2 unit has the lowest MAE.

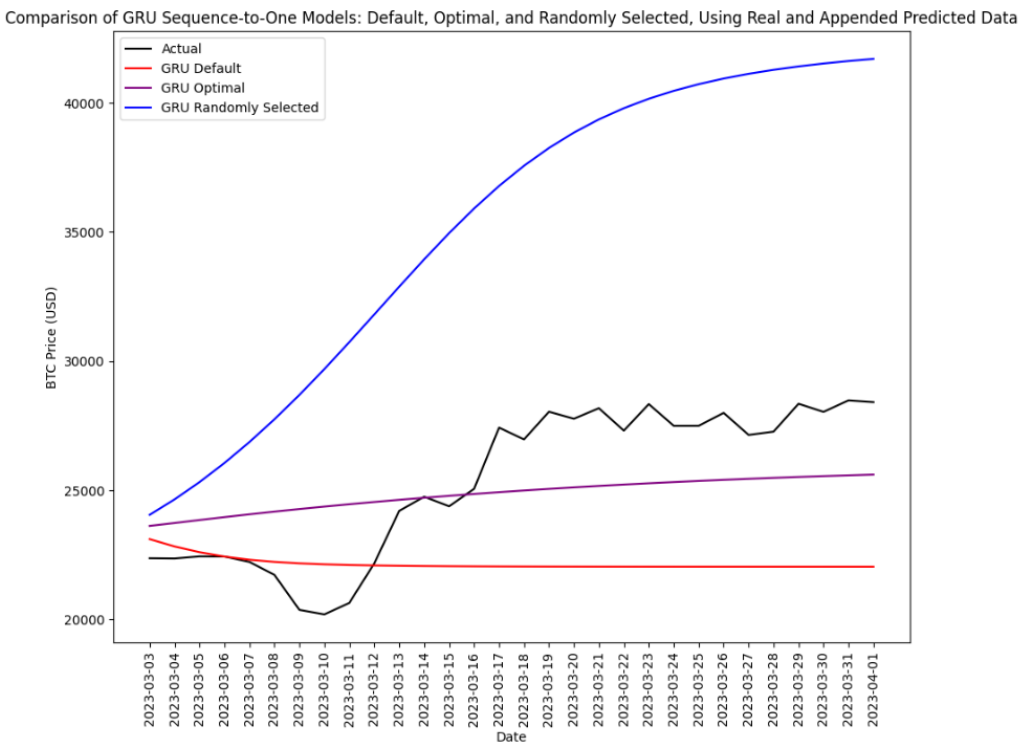


Figure 5.2.2.5 Comparison of GRU Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Walk Forward

	RMSE	MAPE	MAE
GRU Default Sequence-to-One Model Using Real and Appended Predicted Data	4360.300	13.566	3650.011
GRU Optimal Sequence-to-One Model Using Real and Appended Predicted Data	2396.269	8.686	2165.021
GRU Randomly Selected Sequence-to-One Model Using Real and Appended Predicted Data	10466.985	38.150	9833.770

Table 5.2.2.1 Comparison of GRU Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Walk Forward

Figure 5.2.2.5 shows the comparison of GRU Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Real and Appended Predicted Data. Based on the result in Table 5.2.2.1, the optimal model outperforms the other two with the lowest RMSE, MAPE, and MAE, the randomly selected one is the worst among the three.

5.2.3 Experiment 2: Prophet

Sorted RMSE

	changepoint prior scale	seasonality prior scale	RMSE	MAPE	MAE
0	0.005	0.01	2688.342460	10.087775	2441.996108
1	0.005	0.05	2690.340736	10.113666	2448.309979
2	0.005	0.10	2691.807015	10.126247	2452.007965
7	0.010	0.01	2693.598218	10.109193	2447.073906
9	0.010	0.10	2694.810425	10.116570	2448.477173

Sorted MAPE

	changepoint prior scale	seasonality prior scale	RMSE	MAPE	MAE
0	0.005	0.01	2688.342460	10.087775	2441.996108
7	0.010	0.01	2693.598218	10.109193	2447.073906
1	0.005	0.05	2690.340736	10.113666	2448.309979
9	0.010	0.10	2694.810425	10.116570	2448.477173
10	0.010	0.50	2697.710721	10.122687	2449.991403

Sorted MAE

	changepoint prior scale	seasonality prior scale	RMSE	MAPE	MAE
0	0.005	0.01	2688.342460	10.087775	2441.996108
7	0.010	0.01	2693.598218	10.109193	2447.073906
1	0.005	0.05	2690.340736	10.113666	2448.309979
9	0.010	0.10	2694.810425	10.116570	2448.477173
10	0.010	0.50	2697.710721	10.122687	2449.991403

Figure 5.2.3.1 Prophet Sequence-to-One Models Walk Forward Sorted RMSE, MAPE, and MAE

The dataframe from the figure above shows that Prophet Sequence-to-One model with changepoint prior scale 0.005 and seasonality prior scale 0.01 is the optimal model with the lowest RMSE (2688.342), MAPE (10.0878), and MAE (10.0878) among 35 models.

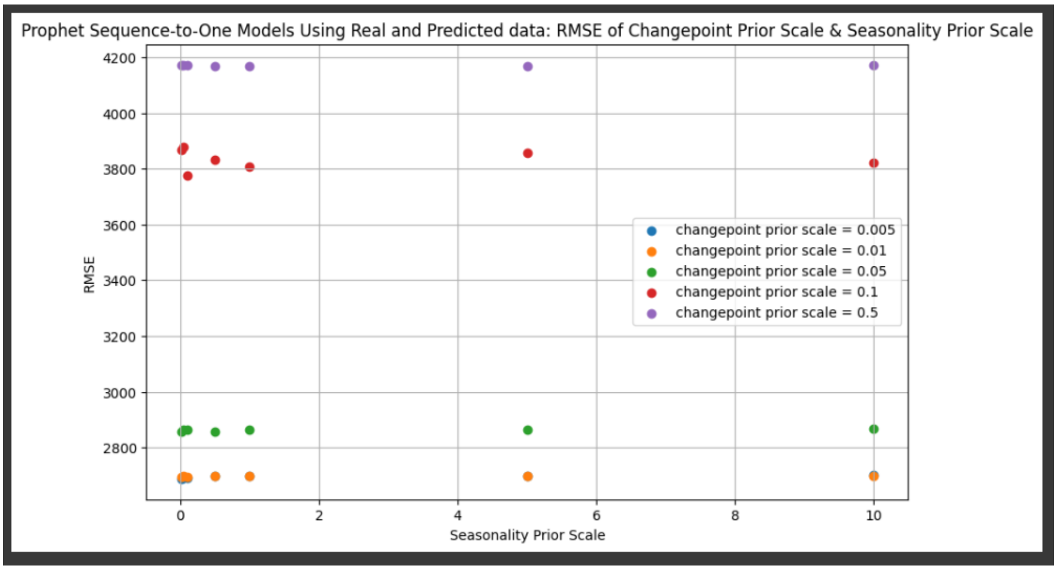


Figure 5.2.3.2 Prophet Sequence-to-One Models Using Walk Forward RMSE Visualization

The figure above shows that Prophet Sequence-to-One model with changepoint prior scale 0.005 and seasonality prior scale 0.01 has the lowest RMSE.

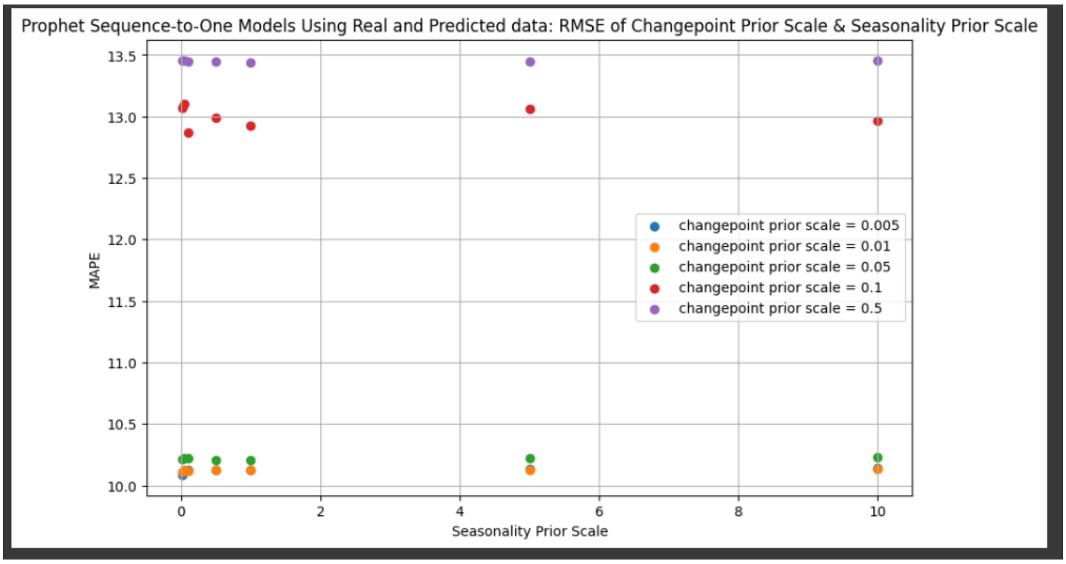


Figure 5.2.3.3 Prophet Sequence-to-One Models Using Walk Forward MAPE Visualization

The figure above shows that Prophet Sequence-to-One model with changepoint prior scale 0.005 and seasonality prior scale 0.01 MAPE.

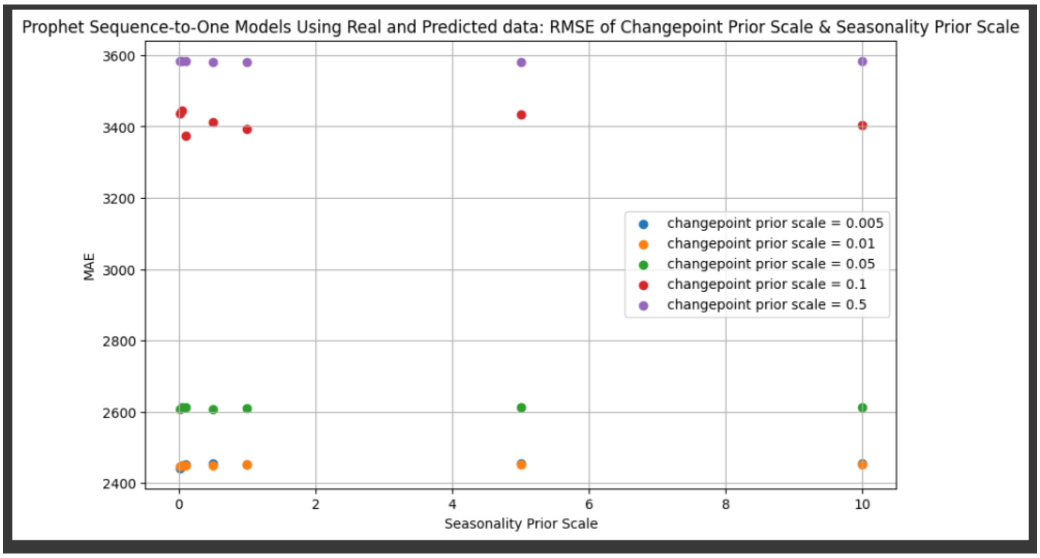


Figure 5.2.3.4 Prophet Sequence-to-One Models Using Walk Forward MAE Visualization

The figure above shows that Prophet Sequence-to-One model with changepoint prior scale 0.005 and seasonality prior scale 0.01 MAE.

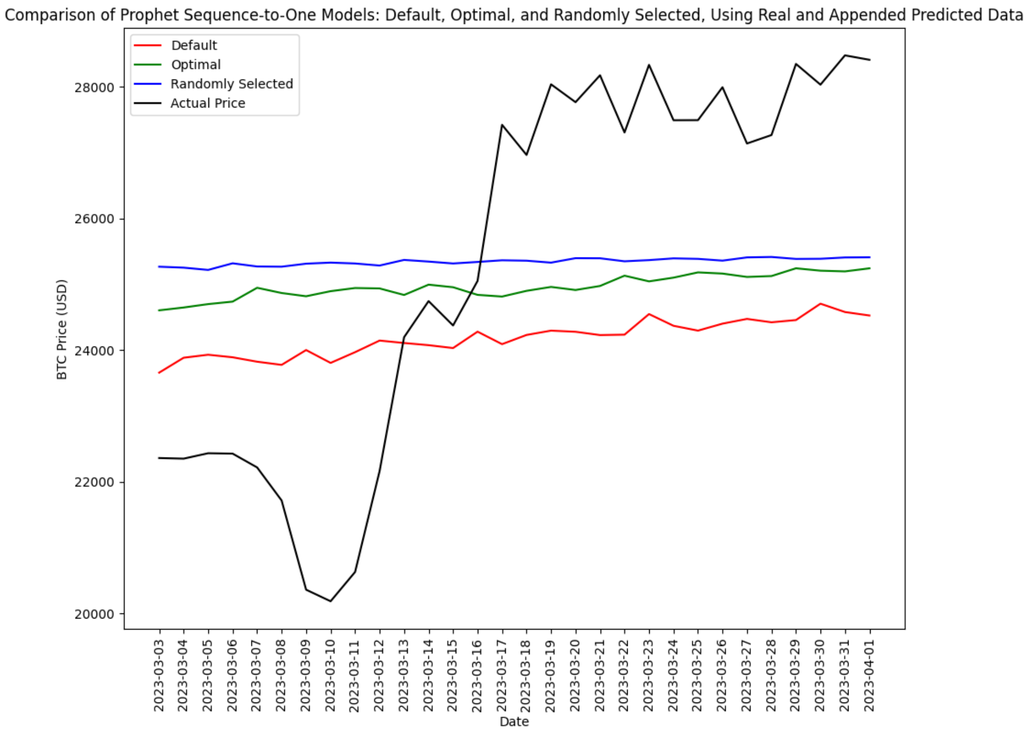


Figure 5.2.3.5 Comparison of Prophet Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Walk Forward

	RMSE	MAPE	MAE
Prophet Default Sequence-to-One Model Using Real and Appended Predicted Data	2865.918	10.228	2613.914
Prophet Optimal Sequence-to-One Model Using Real and Appended Predicted Data	2756.583	10.286	2542.478
Prophet Randomly Selected Sequence-to-One Model Using Real and Appended Predicted Data	2809.249	10.615	2584.270

Table 5.2.3.1 Comparison of Prophet Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Real and Appended Predicted Data

Figure 5.2.3.5 shows the Prophet prediction using walk forward. It indeed looks like the model struggle to predict curvature in time series data.

5.3 Experiment 3: Sequence-to-One Rolling Origin

For the LSTM and GRU, the optimal and a randomly selected models from the LSTM and GRU walk forward was implemented in sequence-to-one forecasting using rolling origin. Meaning, the batch and sliding window consists of the actual historical price data, day (275 until 335) until day (304 until 364), total 30 chunks of data as input to predict 30 days of Bitcoin's closing price.

- The optimal LSTM model: 128 layer 1 units, 128 layer 2 units and 15 epochs
- The randomly selected LSTM model: 128 layer 1 units, 32 layer 2 units and 27 epochs
- The optimal GRU model: 50 layer 1 units, 64 layer 13 units and 15 epochs
- The randomly selected GRU model: 64 layer 1 units, 50 layer 2 units and 14 epochs

5.3.1 Experiment 3: LSTM

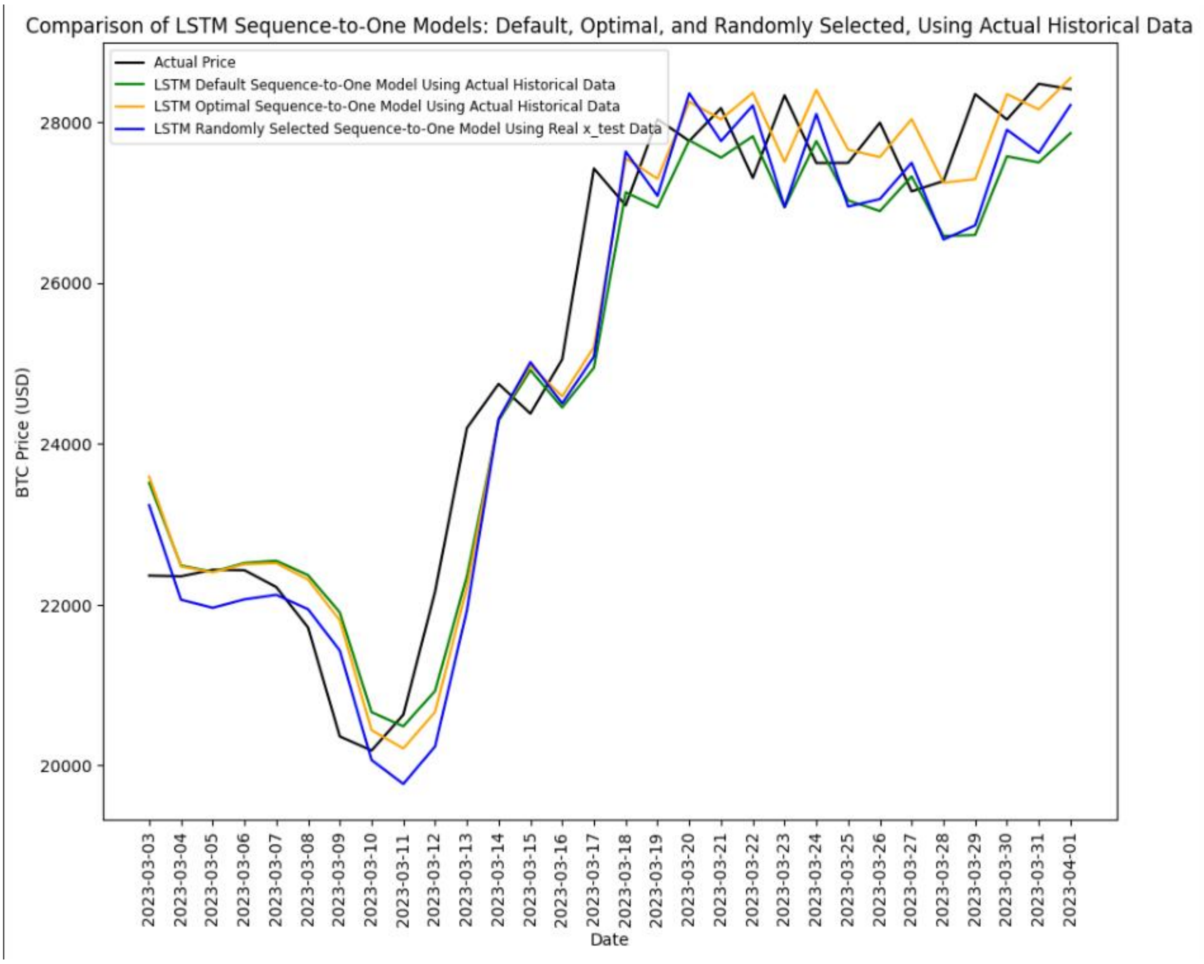


Figure 5.3.1.1 Comparison of LSTM Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Rolling Origin

	RMSE	MAPE	MAE
LSTM Default Sequence-to-One Model Using Actual Historical Data	945.044	2.887	731.345
LSTM Optimal Sequence-to-One Model Using Actual Historical Data	862.353	2.641	658.524
LSTM Randomly Selected Sequence-to-One Model Using Actual Historical Data	977.840	3.092	781.609

Table 5.3.1.1 Comparison of LSTM Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Rolling Origin

Based on Figure 5.3.1.1, surprisingly, the optimal model used in the walk forward performs very well on the rolling origin approach with the lowest RMSE, MAPE and MAE whereas the randomly selected one is still the worst among the three.

5.3.2 Experiment 3: GRU

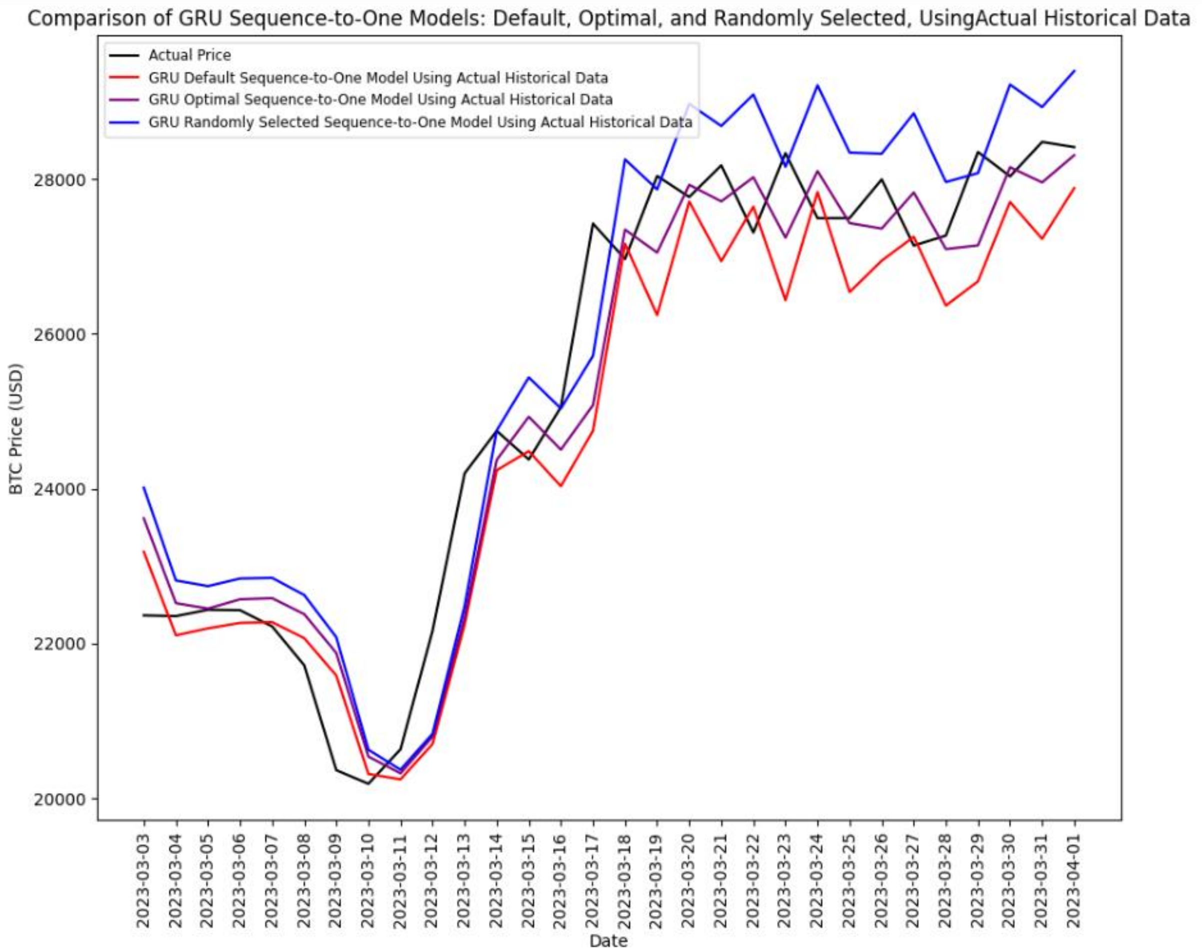


Figure 5.3.2.1 Comparison of GRU Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Rolling Origin

	RMSE	MAPE	MAE
GRU Default Sequence-to-One Model Using Actual Historical Data	1053.200	3.106	800.604
GRU Optimal Sequence-to-One Model Using Actual Historical Data	863.837	2.640	658.239
GRU Randomly Selected Sequence-to-One Model Using Actual Historical Data	1045.584	3.451	864.453

Table 5.3.2.1 Comparison of GRU Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Rolling Origin

Similarly, the optimal GRU model from walk forward is still the optimal model in rolling origin. However, the randomly selected one and the default model configurations are very close to one another.

5.3.3 Experiment 3: Prophet

The optimal and a randomly selected models from the Prophet Architectural Configuration 2 was implemented in sequence-to-one forecasting using the actual data. Meaning, the batch and sliding window consists of the actual historical price data.

- The optimal model: changepoint prior scale 0.005 and seasonality prior scale 0.01
- The randomly selected model: changepoint prior scale 0.5 and seasonality prior scale 10.0

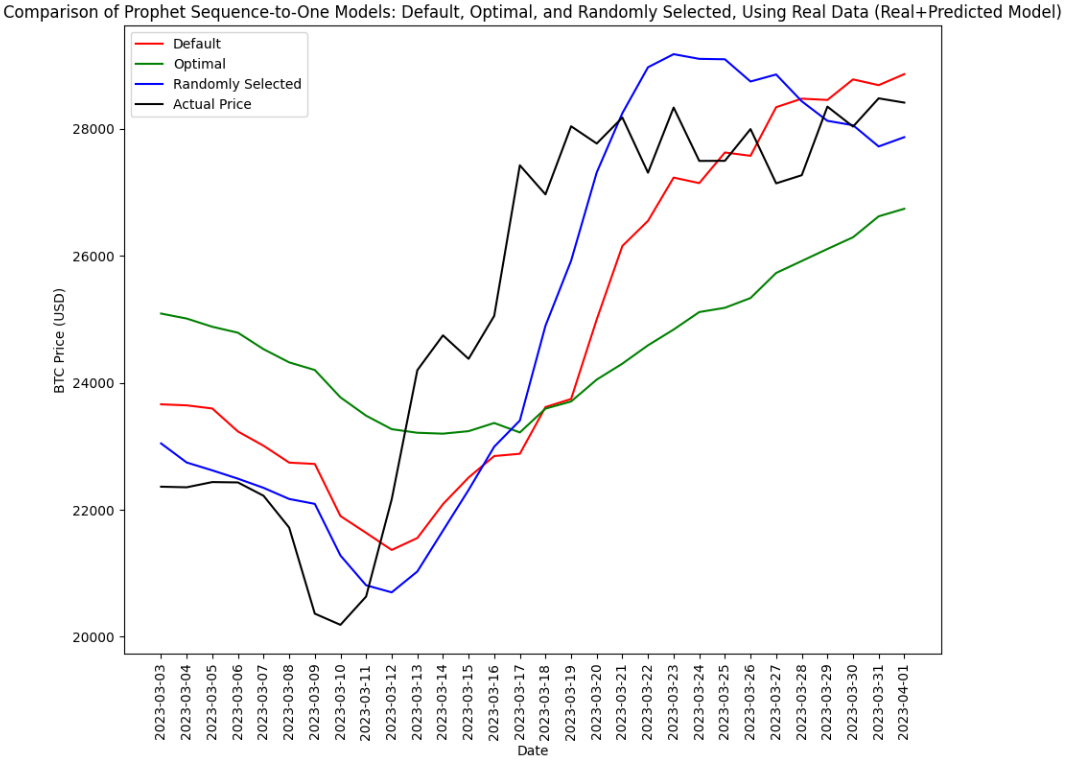


Figure 5.3.3.1 Comparison of Prophet Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Rolling Origin (Sequence-to-One Walk Forward Model)

	RMSE	MAPE	MAE
Prophet Default Sequence-to-One Model Using Actual Historical Data	1892.339	6.026	1508.703
Prophet Optimal Sequence-to-One Model Using Actual Historical Data	2677.689	10.033	2506.259
Prophet Randomly Selected Sequence-to-One Model Using Actual Historical Data	1578.573	4.776	1211.228

Table 5.3.3.1 Comparison of Prophet Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Rolling Origin (Sequence-to-One Walk Forward Model)

Based on Figure 5.3.3.1, and Table 5.3.3.1, the optimal model from walk forward performs poorly on rolling origin with the highest RMSE, MAPE, and MAE among the three, whereas the worst among the 3 models from walk forward performs quite well on the rolling origin.

Since the optimal model from Prophet Sequence-to-One Walk Forward experiment failed to meet the expectation, the experiment was conducted again, using the same approach as Sequence-to-One Walk Forward, instead of appending the predicted values for training, this experiment used the actual data for training, a rolling origin approach to grid search again. Meaning, the batch and sliding window consists of the actual historical price data.

Sorted RMSE

	changepoint prior scale	seasonality prior scale	RMSE	MAPE	MAE
30	0.5	0.1	1578.209355	4.776431	1211.256070
33	0.5	5.0	1578.309800	4.776299	1211.208939
31	0.5	0.5	1578.436314	4.774812	1210.885634
34	0.5	10.0	1578.573357	4.776246	1211.227828
32	0.5	1.0	1578.620143	4.777927	1211.630702

Sorted MAPE

	changepoint prior scale	seasonality prior scale	RMSE	MAPE	MAE
31	0.5	0.5	1578.436314	4.774812	1210.885634
34	0.5	10.0	1578.573357	4.776246	1211.227828
33	0.5	5.0	1578.309800	4.776299	1211.208939
30	0.5	0.1	1578.209355	4.776431	1211.256070
32	0.5	1.0	1578.620143	4.777927	1211.630702

Sorted MAE

	changepoint prior scale	seasonality prior scale	RMSE	MAPE	MAE
31	0.5	0.5	1578.436314	4.774812	1210.885634
33	0.5	5.0	1578.309800	4.776299	1211.208939
34	0.5	10.0	1578.573357	4.776246	1211.227828
30	0.5	0.1	1578.209355	4.776431	1211.256070
32	0.5	1.0	1578.620143	4.777927	1211.630702

Figure 5.3.3.2 Prophet Sequence-to-One Using Rolling Origin Sorted RMSE, MAPE, and MAE

The dataframe from the figure above shows that Prophet Sequence-to-One model with changepoint prior scale 0.5 and seasonality prior scale 0.5 is the optimal model with the third lowest RMSE (1578.436), MAPE (4.775), and MAE (1210.886) among 35 models.

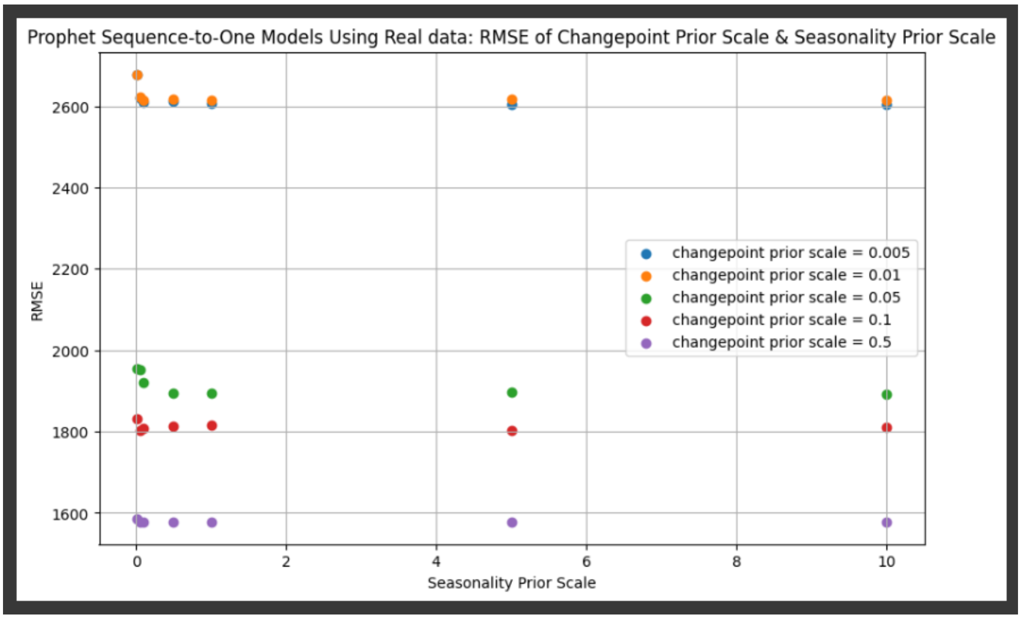


Figure 5.3.3.3 Prophet Sequence-to-One Models Using Rolling Origin RMSE Visualization

The figure above shows that Prophet Sequence-to-One model with changepoint prior scale 0.5 and seasonality prior scale 0.1 has the lowest RMSE whereas the model with changepoint prior scale 0.5 and seasonality prior scale 0.5 has the third lowest RMSE.

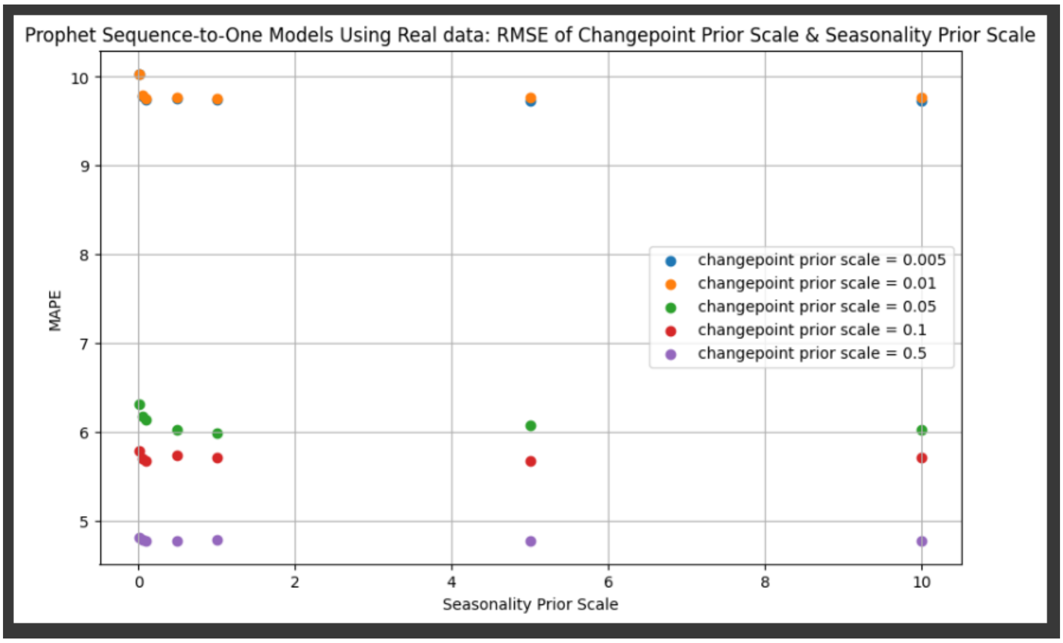


Figure 5.3.3.4 Prophet Sequence-to-One Models Using Rolling Origin MAPE Visualization

The figure above shows that Prophet Sequence-to-One model with changepoint prior scale 0.5 and seasonality prior scale 0.5 has the lowest MAPE.

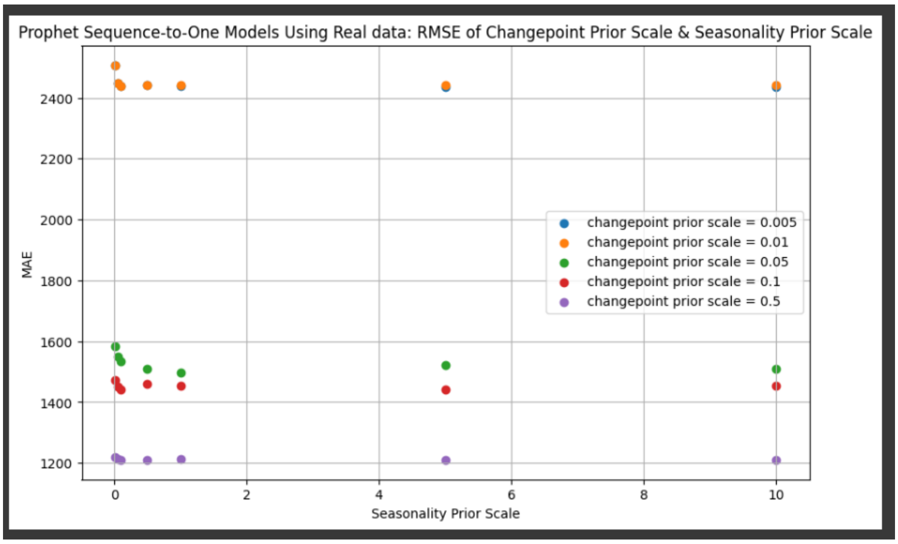


Figure 5.3.3.5 Prophet Sequence-to-One Models Using Rolling Origin MAE Visualization

The figure above shows that Prophet Sequence-to-One model with changepoint prior scale 0.5 and seasonality prior scale 0.5 has the lowest MAE.

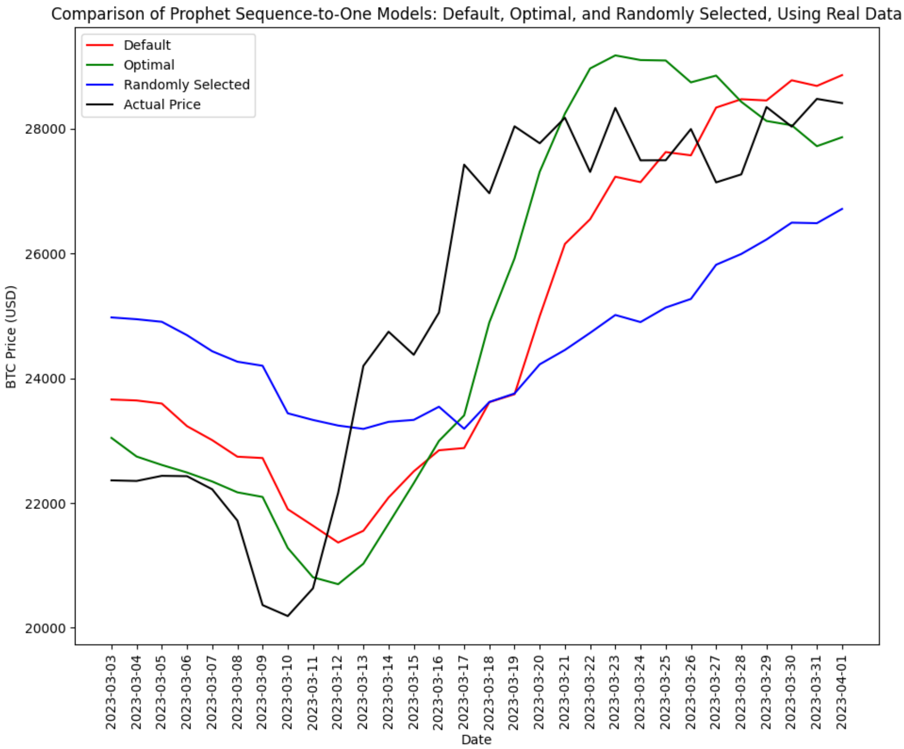


Figure 5.3.3.6 Comparison of Prophet Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Rolling Origin

	RMSE	MAPE	MAE
Prophet Default Sequence-to-One Model Using Actual Historical Data	1892.339	6.026	1508.703
Prophet Optimal Sequence-to-One Model Using Actual Historical Data	1578.436	4.775	1210.886
Prophet Randomly Selected Sequence-to-One Model Using Actual Historical Data	2612.316	9.759	2440.751

Table 5.3.3.2 Comparison of Prophet Sequence-to-One Models: Default, Optimal, and Randomly Selected, Using Rolling Origin

Based on Figure 5.3.3.6, the optimal model (seasonality prior scale 0.5 and changepoint prior scale 0.5) is the best performing out of the 3, with the lowest RMSE, MAPE, and MAE.

5.4 Comparison and Discussion

A total of 2 architectural configurations (Sequence-to-Sequence and Sequence-to-One) and 3 experiments were conducted. The following compares the performance of the optimal models obtained from grid search.

5.4.1 Experiment 1: Sequence-to-Sequence

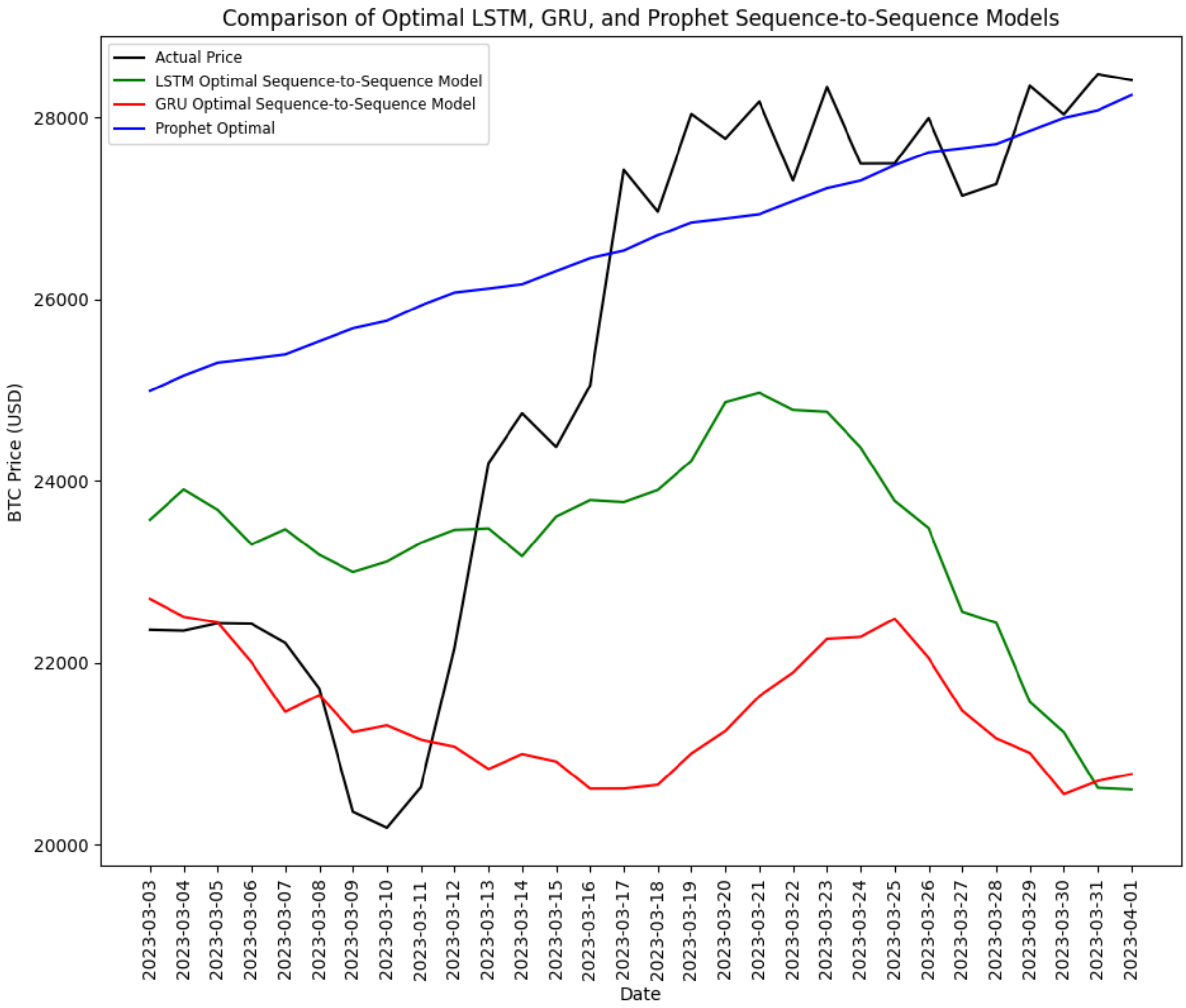


Figure 5.4.1.1 Comparison of Sequence-to-Sequence Models

	RMSE	MAPE	MAE
LSTM	3730.464	12.204	3182.361
GRU	4945.611	15.388	4135.137
Prophet	2431.102	7.884	1781.180

Table 5.4.1.1 Comparison of Sequence-to-Sequence Models

Based on Figure 5.4.1.1, and Table 5.4.1.1 above, in Sequence-to-Sequence architecture configuration, the Prophet model (depicted in purple) achieves the lowest RMSE, MAPE and MAE values compared to LSTM (green) and GRU (red). GRU has the highest RMSE,MAPE and MAE among the three models. However, it is important to note that having low RMSE, MAPE, and MAE does not necessarily mean the model prediction is good. In this experiment, the Prophet model is struggling to predict curvature and fluctuations in time series data. On the other hand, both LSTM and GRU models managed to forecast a decreasing trend in the first 6 days followed by an increase even though the precise price is far from accurate. In this case, Prophet's lower error metrics do not indicate its superiority over the LSTM and GRU models.

5.4.2 Experiment 2 Sequence-to-One Walk Forward

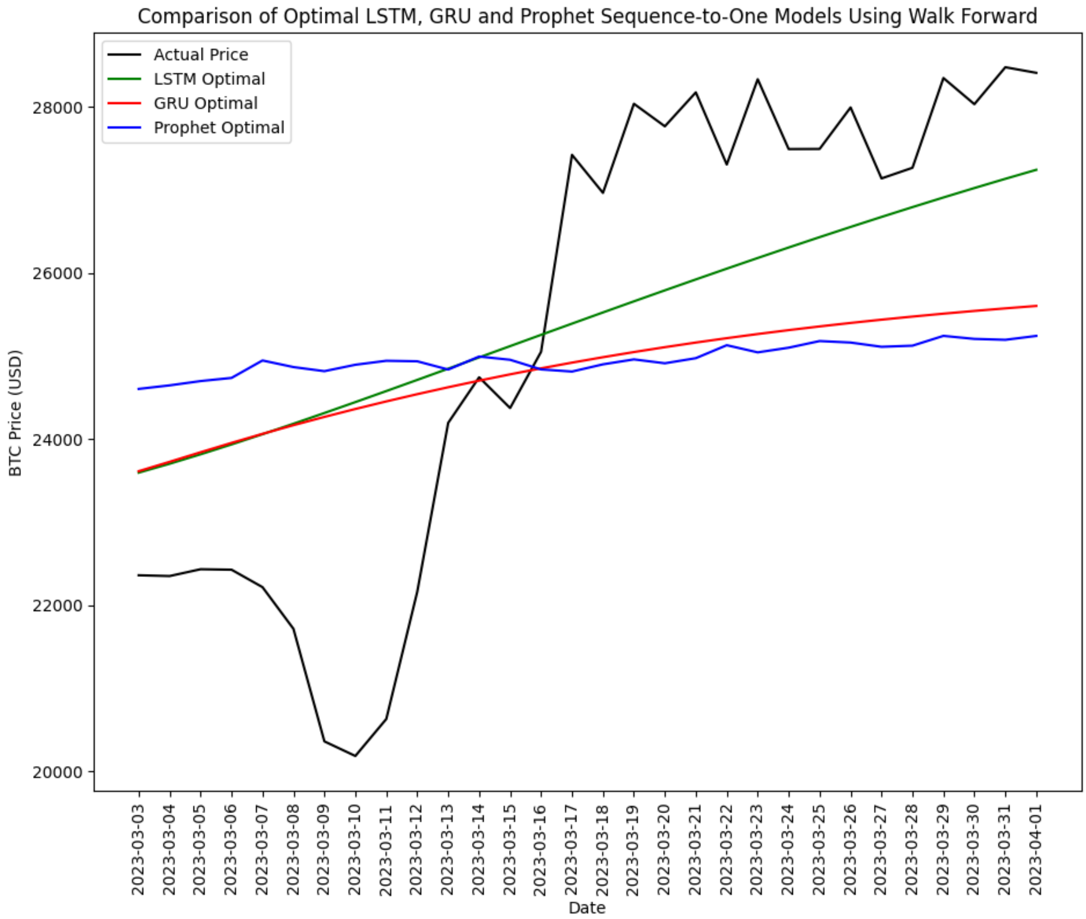


Figure 5.4.2.1 Comparison of Sequence-to-One Walk Forward Models

	RMSE	MAPE	MAE
LSTM	1934.700	6.853	1646.739
GRU	2396.269	8.686	2165.021
Prophet	2756.583	10.286	2542.478

Table 5.4.2.1 Comparison of Sequence-to-One Walk Forward Models

Based on the data presented in Figure 5.4.2.1, and Table 5.4.2.1 above, in Sequence-to-One Walk Forward architecture configuration prediction by appending prediction into input batch out of sample prediction, LSTM achieves the lowest RMSE, MAPE and MAE values while Prophet has the highest among the three models. Again, this experiment shows that Prophet is poor at predicting Bitcoin price. LSTM and GRU struggle to forecast the decrease trend in the short term but show an upward trend in longer term. However, this method of evaluation, where the initial predicted data point is significantly inaccurate in the first place and is then appended to the input batch for subsequent predictions, it introduces noise into the model performance. This process accumulates errors, eventually causing the predictions worse over time. However, walk forward simulates real time scenarios. It reflects how a model would perform when used in real time. Predicting fluctuations in walk forward time series data can be challenging, especially when Bitcoin price is highly fluctuated, highly volatile, and highly unstable.

5.4.3 Experiment 3 Sequence-to-One Rolling Origin

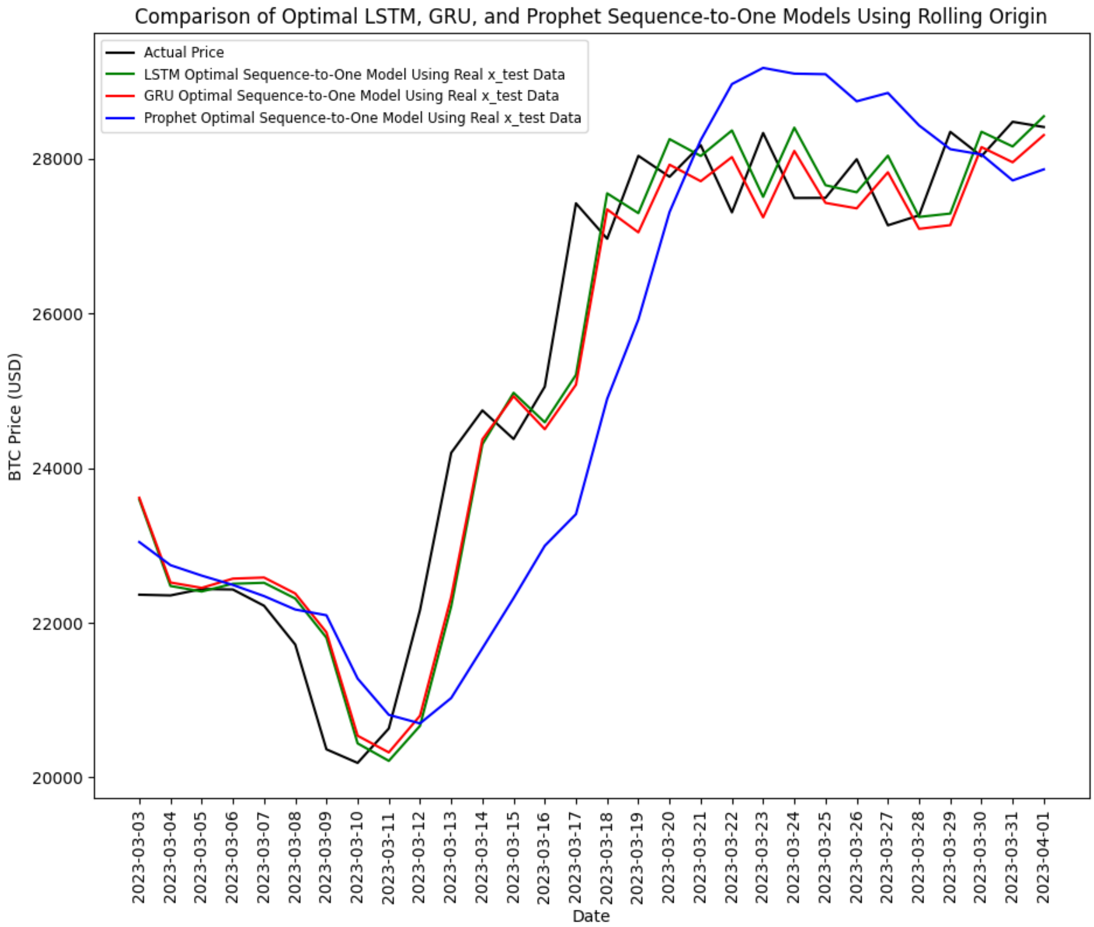


Figure 5.4.1.3 Comparison of Sequence-to-One Rolling Origin Models

	RMSE	MAPE	MAE
LSTM	862.353	2.641	658.524
GRU	863.837	2.640	658.239
Prophet	1578.436	4.775	1210.886

Table 5.4.1.3 Comparison of Sequence-to-One Rolling Origin Models

Figure 5.4.1.3 and Table 5.4.1.3 shows that Prophet has the worst performance, whereas LSTM and GRU is very close to each other in Sequence-to-One Rolling Original models using actual historical closing price data for prediction. However, this approach only works when the Bitcoin closing price is known.

5.4.4 Prophet Optimal Walk Forward Model Poor Performance Investigation

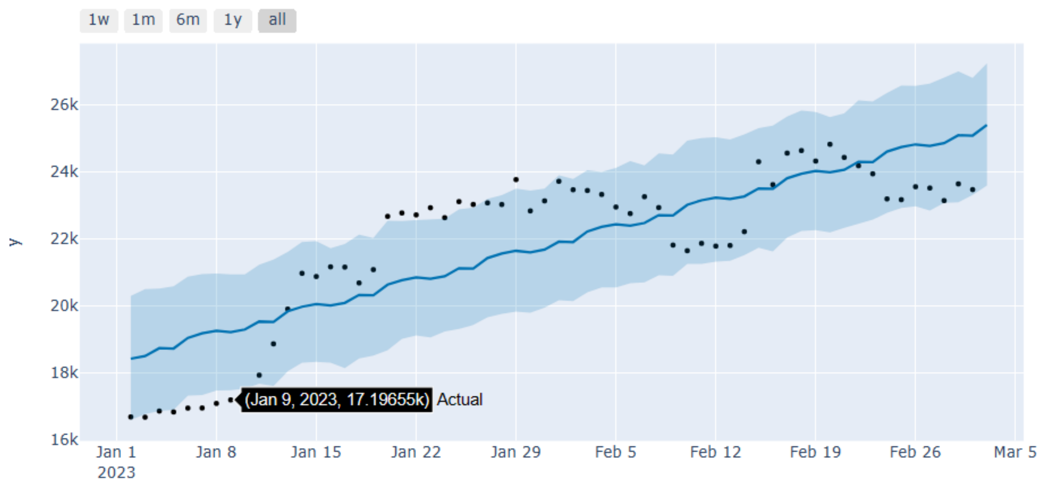


Figure 5.4.4.1 Changepoint prior scale 0.005

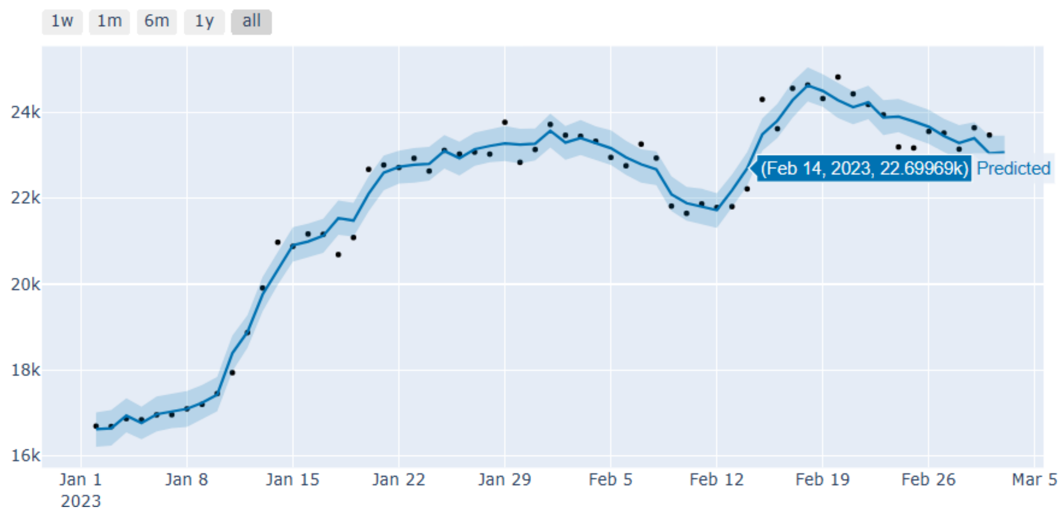


Figure 5.4.4.2 Changepoint prior scale 0.5

Upon further investigation into the optimal model of Prophet in Sequence-to-One walk forward prediction performs poorly in rolling origin is because the changepoint plays a crucial role in determining the accuracy of forecast. Figure 5.4.4.1 and Figure 5.4.4.2 illustrate the predictions and actual Bitcoin price in the input batch for Sequence-to-One prediction. The actual Bitcoin prices are represented by black dots and the blue line represents the model's predictions. In Figure 5.4.4.1 the changepoint prior scale is 0.005, and the model has difficulty capturing the pattern in the Bitcoin price. The initial predicted data point is also highly inaccurate. Continuing with the walk forward method, the model introduces noises into the predictions, bringing them closer to the actual price.

However, this noise does not imply good learning by the model. In contrast, Figure 5.4.4.2's model with 0.5 changepoint prior has effectively learned the pattern of the Bitcoin price, making it superior. That explains why the optimal walk forward model performs poorly on the rolling origin approach.

5.4.5 Experiments Summary

LSTM stands out as the optimal choice among the three, in Sequence-to-Sequence, Sequence-to-One walk forward or Sequence-to-One rolling origin Bitcoin closing price prediction. GRU comes in second place. The problem with Prophet is that the model is struggling to predict fluctuations and curvature in this highly unstable Bitcoin price prediction task.

CHAPTER 6

Conclusion

Due to Bitcoin's market capitalization, value, volatility, and its decentralized properties. Bitcoin has caught the attention of many people, from ordinary citizen and investors to machine learning researchers and data scientist. Past studies mostly used traditional time series and deep learning algorithms for Bitcoin price prediction. This research paper proposed 3 machine learning algorithms with 2 different architectural configurations, namely Sequence-to-Sequence and Sequence-to-One with 2 different evaluation methods based on past 1 year of Bitcoin dataset and the models are evaluated using RMSE, MAPE, and MAE. A total of 3 experiments (Sequence-to-Sequence, Sequence-to-One walk forward or Sequence-to-One rolling origin) was conducted. In addition, hyperparameter tuning was performed to improve prediction accuracy.

LSTM stands out as the optimal choice for Bitcoin price prediction among the three models and experiments, GRU comes in second place. The problem with Prophet is that the model struggles to predict fluctuations and curvature in this highly unstable Bitcoin price prediction task. Having lower error metrics does not imply that the model is good for prediction.

It is important to note that, predicting highly volatile and unstable time series data like Bitcoin price is undoubtedly a very challenging task. This research has provided valuable insights into the different architectural configurations of algorithms. The limitation of this research is that it relies on historical Bitcoin data. It may not fully capture the rapidly changing nature of the market in real time settings. Relying solely on the closing price for training model may not be sufficient in forecasting with high precision and accuracy. Using only the closing price gives limited information about the behaviour of the financial market. Financial markets are influenced by a wide range of factors, from economy to news, social media, supply and demand, government regulations and even politics. Considering the mentioned factors in training model may lead to a potential performance improvement. Future research should consider exploring external factors and implementing the models to continuously gather and process real time Bitcoin price data.

REFERENCES

[1] Amin Azari, “Bitcoin Price Prediction: An ARIMA Approach”, Cornell University, 2019.

[2] I Made Wirawan et. al. “Short Term Prediction on Bitcoin Price Using ARIMA Method”, 2019.

[3] S. Roy, S. Nanjiba and A. Chakrabarty, "Bitcoin Price Forecasting Using Time Series Analysis," 2018 21st International Conference of Computer and Information Technology (ICCIT), 2018, pp. 1-5, doi: 10.1109/ICCITECHN.2018.8631923.

[4] Anshul Saxena, "Predicting Bitcoin Price Using LSTM And Compare Its Predictability With Arima Model", Christ University, 2018.

[5] S. McNally, J. Roche and S. Caton, "Predicting the Price of Bitcoin Using Machine Learning," 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), 2018, pp. 339-343, doi: 10.1109/PDP2018.2018.00060.

[6] Muniye, Temesgen & Rout, Minakhi & Mohanty, Lipika & Satapathy, Suresh, "Bitcoin Price Prediction and Analysis Using Deep Learning Models ", 2020.

[7] K. Rathan, S. V. Sai and T. S. Manikanta, "Crypto-Currency price prediction using Decision Tree and Regression techniques," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 190-194, doi: 10.1109/ICOEI.2019.8862585.

[8] Isil Yenidogan et. al. “Bitcoin Forecasting Using ARIMA and Prophet”, 2018.

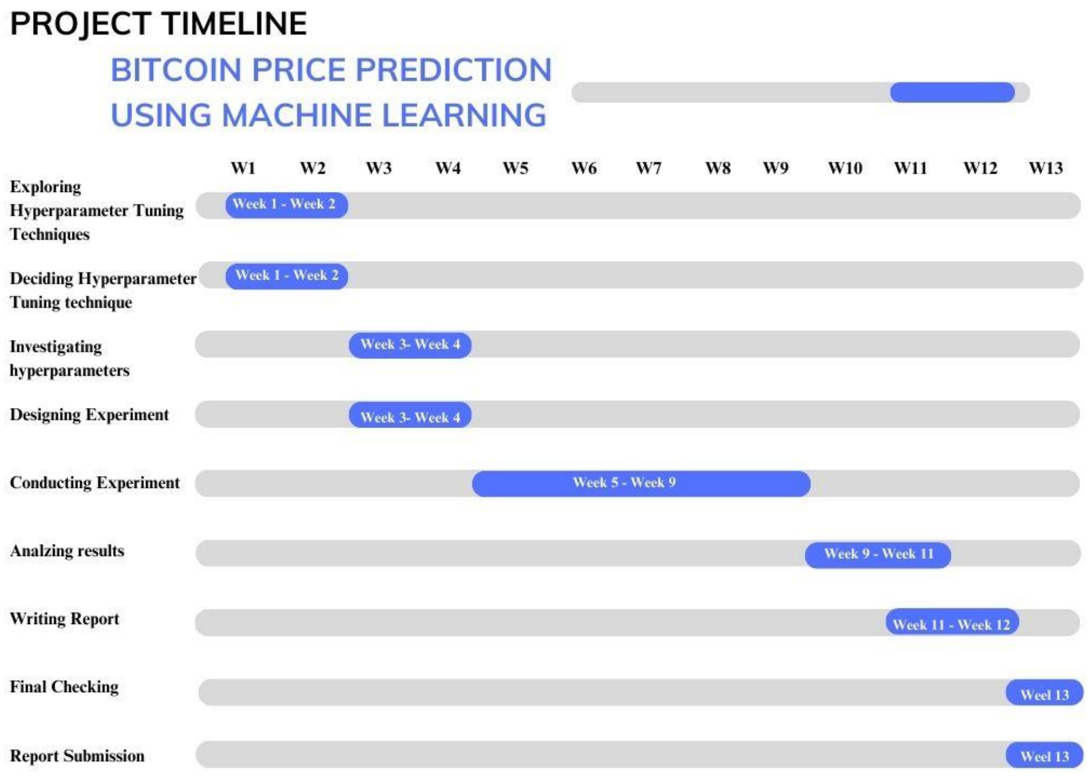
[9] Colah, Understanding LSTM Networks, 2015.

[10] Micheal Phi, Illustrated Guide to LSTM’s and GRU’s, 2018.

[11] Rahulhegde97, Time Series Analysis Using Facebook Prophet.

APPENDIX

Gantt Chart



FINAL YEAR PROJECT BI-WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 2**
Student Name & ID: Tang Jian Yang (20ACB05721)	
Supervisor: Dr Tong Dong Ling	
Project Title: Bitcoin Price Prediction Using Machine Learning	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Explored hyperparameter tuning techniques.
- Decided to use parameter grid and grid search.
- Installed and imported necessary packages and libraries.

2. WORK TO BE DONE

- Investigate and research the hyperparameters of LSTM. GRU and Prophet.
- Designing experiments

3. PROBLEMS ENCOUNTERED

Jupyter notebook could not installed the required packages and libraries.

4. SELF EVALUATION OF THE PROGRESS

Exploring hyperparameter tuning techniques is a valuable initial step in machine learning projects.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BI-WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 4**
Student Name & ID: Tang Jian Yang (20ACB05721)	
Supervisor: Dr Tong Dong Ling	
Project Title: Bitcoin Price Prediction Using Machine Learning	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Found a Jupyter notebook alternative – Google Colab (Web based)
- Gained knowledge and information about the hyperparameters.
- Decided to use Sequence-to-Sequence architecture for experiment 1 and Sequence-to-One architecture for experiment 2. Both uses walk forward validation method.

2. WORK TO BE DONE

- Begin experiment and testing various combination of hyperparameters using grid search on LSTM, GRU, and Prophet.

3. PROBLEMS ENCOUNTERED

Faced challenges while building the Sequence-to-Sequence architecture, but managed to resolve it.

4. SELF EVALUATION OF THE PROGRESS

Studying and understanding the functions of hyperparameter is crucial. Designed experiments. In the right direction.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BI-WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 6
Student Name & ID: Tang Jian Yang (20ACB05721)	
Supervisor: Dr Tong Dong Ling	
Project Title: Bitcoin Price Prediction Using Machine Learning	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Completed the experiments.
- Completed and determined the optimal model of each algorithm for each experiment.

2. WORK TO BE DONE

- Conduct a new experiment – try applying the optimal model of each algorithm of experiment 2 on rolling origin approach.

1. PROBLEMS ENCOUNTERED

None.

2. SELF EVALUATION OF THE PROGRESS

Experiments completed but keep exploring new things. Good learning behavior.

Tong

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BI-WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 8**
Student Name & ID: Tang Jian Yang (20ACB05721)	
Supervisor: Dr Tong Dong Ling	
Project Title: Bitcoin Price Prediction Using Machine Learning	

3. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Completed LSTM and GRU experiment 3.

4. WORK TO BE DONE

- Redo Prophet experiment 3 again using rolling origin approach.
- Investigate the root cause of it.

3. PROBLEMS ENCOUNTERED

- The optimal model of LSTM and GRU from experiment 2 works well on experiment 3 but the optimal experiment 2 Prophet model performs poorly on experiment 3.

4. SELF EVALUATION OF THE PROGRESS

I should start writing report.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BI-WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 10**
Student Name & ID: Tang Jian Yang (20ACB05721)	
Supervisor: Dr Tong Dong Ling	
Project Title: Bitcoin Price Prediction Using Machine Learning	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Started writing report.
- Completed Prophet experiment 3.
- Investigated the cause of Prophet experiment 2 optimal model's poor performance on experiment 3.

2. WORK TO BE DONE

- Finish writing the report.

3. PROBLEMS ENCOUNTERED

None.

4. SELF EVALUATION OF THE PROGRESS

All experiments done. Focus on writing the report.

Tong

Supervisor's signature

Student's signature

POSTER


BITCOIN PRICE PREDICTION USING MACHINE LEARNING
Introduction
Bitcoin caught the attention of many people. Machine learning can be used to predict Bitcoin price
Objective
To find the optimal model among LSTM, GRU, and Prophet of different architectural configurations
Methods
• Train on past Bitcoin dataset, 335 days training data and 30 days testing data
• Sliding window
• Grid search hyperparameter tuning
• Evaluated using RMSE, MAPE and MAE
Experiments
• Sequence-to-Sequence Walk Forward
• Sequence-to-One Walk Forward
• Sequence-to-One Rolling Origin
Results
• LSTM achieves the best performance
• GRU comes in second place
• Prophet struggles to predict fluctuations and curvature
• Having low error metrics does not imply the prediction is good
Novelty and Significance
• Prophet algorithm sliding window approach prediction
• Provide broader range of options to choose for forecasting Bitcoin price
UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Faculty of Information Communication and Technology
Project Researcher: Tang Jian Yang
Project Supervisor: Dr Tong Dong Ling

PLAGIARISM CHECK RESULT

Bitcoin price prediction using machine learning

ORIGINALITY REPORT

9%	6%	7%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	medium.com Internet Source	1%
2	Sean McNally, Jason Roche, Simon Caton. "Predicting the Price of Bitcoin Using Machine Learning", 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), 2018 Publication	1%
3	www.mdpi.com Internet Source	1%
4	Submitted to Liverpool John Moores University Student Paper	1%
5	"Communication Software and Networks", Springer Science and Business Media LLC, 2021 Publication	<1%
6	link.springer.com Internet Source	<1%

Universiti Tunku Abdul Rahman			
Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Tang Jian Yang
ID Number(s)	001031-08-0379
Programme / Course	Bachelor of Computer Science (HONOURS) / FYP 2
Title of Final Year Project	Bitcoin Price Prediction Using Machine Learning

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 9 % **Similarity by source** Internet Sources: 6 % Publications: 7 % Student Papers: 4 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required, and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor	Signature of Co-Supervisor
Name: Tong Dong Ling	Name:
Date: 15 Sep 2023	Date:

FYP 2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP1 THESIS SUBMISSION

Student ID	20ACB05721
Student Name	Tang Jian Yang
Supervisor Name	Tong Dong Ling

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgment
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

.

I, the author, have checked and confirmed all the items listed in the table are included in my report. _________________________ (Signature of Student) Date: 14/9/2023