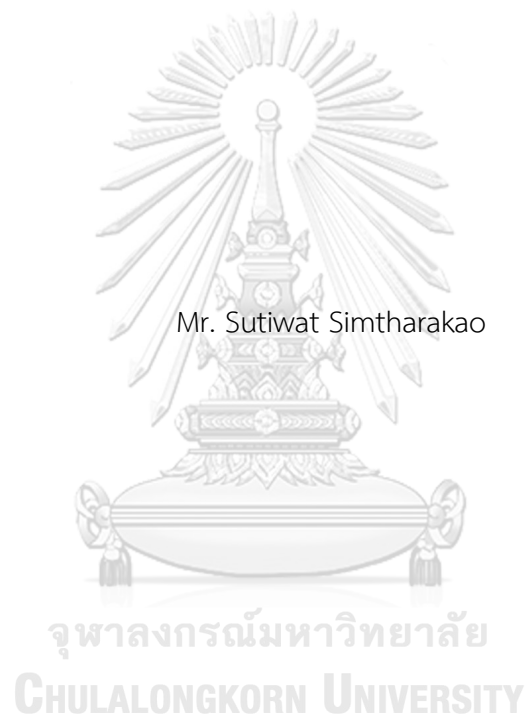


Bitcoin Candlestick Price Prediction with Recurrent Neural Network



Mr. Sutiwat Simtharakao

A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Engineering in Industrial Engineering

Department of Industrial Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2022

Copyright of Chulalongkorn University

การทำนายราคาแท่งเทียนของบิทคอยน์ด้วยโครงข่ายประสาทเทียมแบบวนซ้ำ



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2565

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย



สุทิวส์ สิมธาราแก้ว : การทำนายราคาแท่งเทียนของบิทคอยน์ด้วยโครงข่ายประสาทเทียมแบบวนซ้ำ.  
(Bitcoin Candlestick Price Prediction with Recurrent Neural Network) อ.ที่ปรึกษาหลัก :  
รศ. ดร.ดาริชา สุธีวงศ์

บิทคอยน์เป็นหนึ่งในสินทรัพย์ที่มีผลตอบแทนและความเสี่ยงสูง การทำนายราคาแท่งเทียนของบิทคอยน์จะช่วยให้นักลงทุนมีข้อมูลเพิ่มเติมในการตัดสินใจ งานวิจัยนี้มีวัตถุประสงค์เพื่อสร้างแบบจำลองโครงข่ายประสาทเทียมเพื่อทำนายราคาแท่งเทียนของบิทคอยน์ในวันถัดไป และเพื่อศึกษาวิธีการปรับปรุงประสิทธิภาพในการทำนายของแบบจำลอง โดยใช้การแปลงคุณสมบัติของข้อมูล กล่าวคือการปรับช่วงของข้อมูล โดยงานวิจัยนี้ใช้แบบจำลองโครงข่ายประสาทเทียมแบบวนซ้ำสองชนิด คือ หน่วยความจำระยะสั้นแบบยาว (LSTM) และโครงข่ายประตูกลับ (GRU) เพื่อที่จะพยากรณ์ราคาแท่งเทียนรายวันของบิทคอยน์ สำหรับการปรับปรุงสมรรถนะของแบบจำลอง งานวิจัยนี้เปรียบเทียบวิธีการปรับช่วงของข้อมูลสองวิธีคือ การปรับช่วงของข้อมูลโดยใช้คุณสมบัติของชุดข้อมูลฝึกสอน (whole set normalization) และการปรับช่วงของข้อมูลแบบเคลื่อนที่ (sliding window normalization) ซึ่งงานวิจัยนี้ศึกษาเทคนิคการปรับช่วงข้อมูล 3 ชนิด คือ การปรับช่วงข้อมูลด้วยค่ามาตรฐาน (z-score), การปรับช่วงข้อมูลด้วยค่ามากที่สุดและค่าน้อยสุด (min-max), และการปรับช่วงข้อมูลแบบการเปลี่ยนแปลงสัมพัทธ์ (relative change) นอกจากนี้งานวิจัยนี้ยังเปรียบเทียบวิธีการสร้างแท่งเทียน 2 วิธี คือ การใช้ราคาแท่งเทียน (OHLC) และการใช้ไส้เทียน (CULR) ในการทำนาย ทั้งนี้แบบจำลองในงานวิจัยนี้จะใช้ราคาแท่งเทียนในอดีตหลายวันเพื่อจะพยากรณ์ราคาแท่งเทียนของวันถัดไป จากการทดลองพบว่าแบบจำลองที่มีประสิทธิภาพในการพยากรณ์ที่ดีที่สุดนั้นคือ แบบจำลองโครงข่ายประตูกลับ ร่วมกับการปรับช่วงข้อมูลด้วยค่ามาตรฐานแบบเคลื่อนที่ และการสร้างแท่งเทียนด้วยการใช้ราคาแท่งเทียน โดยมีค่าเฉลี่ยร้อยละของความคลาดเคลื่อนสัมบูรณ์ (MAPE) ที่ต่ำที่สุดคือ 1.95% และมีค่ารากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง (RMSE) ที่ต่ำที่สุดคือ 767.71 นอกจากนี้ยังพบว่าการใช้การปรับช่วงของข้อมูลแบบเคลื่อนที่มีค่าความคลาดเคลื่อนที่ต่ำกว่าการปรับช่วงของข้อมูลโดยใช้คุณสมบัติของชุดข้อมูลฝึกสอนในแบบจำลองทั้งสองชนิด สำหรับวิธีการสร้างแท่งเทียนพบว่าไม่มีความแตกต่างกันอย่างมีนัยสำคัญในแง่ของค่าความคลาดเคลื่อนของการพยากรณ์ และการทำนายความแม่นยำในการทำนายทิศทาง แต่พบว่าการใช้ราคาแท่งเทียนมีประสิทธิภาพดีกว่าเล็กน้อย

สาขาวิชา วิศวกรรมอุตสาหการ  
ปีการศึกษา 2565

ลายมือชื่อนิสิต .....  
ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

# # 6470364521 : MAJOR INDUSTRIAL ENGINEERING

KEYWORD: Bitcoin, Neural networks, Cryptocurrency, Forecasting, Price Prediction,  
Candlestick, Normalization, Recurrent Neural Network, LSTM, GRU

Sutiwat Simtharakao : Bitcoin Candlestick Price Prediction with Recurrent Neural  
Network. Advisor: Assoc. Prof. DARICHA SUTIVONG, Ph.D.

Bitcoin is a high-risk asset with a potentially high return. Predicting Bitcoin candlestick, i.e., open, high, low, and close (OHLC) prices, can help investors make trading decisions. The objective of this study is to develop a neural network model to predict the candlestick prices of Bitcoin for the next period. Additionally, this study investigates methods to enhance the model's forecasting performance by feature transformations, specifically data normalization. This study employs two neural network algorithms, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), to forecast daily Bitcoin OHLC prices. To enhance the model's performance, we compare sliding window normalization with whole set normalization techniques. The normalization techniques investigated for both whole set and sliding window data include z-score normalization, min-max normalization, and relative change normalization. Furthermore, this study compares two candlestick prediction methods, namely using OHLC prices and using candle wick (CULR) to predict OHLC prices. The models use historical OHLC prices over several days to predict the next day's OHLC prices. The results indicate that the best-performing model is the OHLC method using GRU algorithm with sliding window z-score normalization, which achieves an MAPE of 1.95% and an RMSE of 767.71. Moreover, the sliding window normalization generally outperforms the whole set normalization for both LSTM and GRU models in terms of RMSE and MAPE. Regarding the candlestick prediction methods, there was no significant difference in their performance in terms of accuracy and forecasting error. However, our results suggest that the OHLC method performs slightly better than the CULR method.

Field of Study: Industrial Engineering

Student's Signature .....

Academic Year: 2022

Advisor's Signature .....

## ACKNOWLEDGEMENTS

I am deeply grateful to Assoc. Prof. Daricha Sutivong, Ph.D., for serving as my thesis advisor and providing invaluable guidance, insightful knowledge, and constructive feedback that were essential in successfully completing this thesis. Her expertise and dedication have been a source of inspiration and motivation to pursue academic excellence. I would like to extend my sincere appreciation to my colleagues and friends whose unwavering support, encouragement, and motivation were invaluable in every step of this research journey. My gratitude also goes to The Regional Center of Robotics Technology and its members for their invaluable contribution in providing state-of-the-art equipment, facilities, and resources that were crucial in conducting this research. Finally, I would like to express my heartfelt thanks to my parents and all those who have encouraged and supported me throughout my academic and personal life. Your unwavering love, support, and motivation have been instrumental in shaping my character and enabling me to achieve my academic goals.

Sutiwat Simtharakao



จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

## TABLE OF CONTENTS

	Page
.....	iii
ABSTRACT (THAI) .....	iii
.....	iv
ABSTRACT (ENGLISH) .....	iv
ACKNOWLEDGEMENTS .....	v
TABLE OF CONTENTS .....	vi
LIST OF TABLES .....	x
LIST OF FIGURES.....	xiii
Chapter 1: Introduction.....	1
1.1 Background.....	1
1.2 Objectives of the study .....	2
1.3 Scope of the study .....	2
1.4 Expected outcomes of the study.....	2
1.5 Expected benefits of the study .....	2
Chapter 2: Related theories and literature review .....	3
2.1 Related theories .....	3
2.1.1 Neural network.....	3
2.1.2 Recurrent neural network (RNN).....	3
2.1.2.1 Simple RNN .....	3
2.1.2.2 Long short-term memory (LSTM).....	4
2.1.2.3 Gated recurrent unit (GRU).....	6
2.1.3 Forecasting performance metric.....	7
2.1.3.1 Mean absolute error (MAE).....	7
2.1.3.2 Mean absolute percentage error (MAPE) .....	7

2.1.3.3	Mean square error (MSE).....	7
2.1.3.4	Root mean square error (RMSE).....	8
2.1.4	Classification metrics.....	8
2.1.4.1	Accuracy .....	8
2.1.4.2	Precision .....	8
2.1.4.3	Recall .....	8
2.1.4.4	F1-Score .....	9
2.1.5	Data normalization and scaling techniques.....	9
2.1.5.1	Z-Score normalization .....	9
2.1.5.2	Min-Max normalization.....	9
2.1.5.3	Relative change normalization.....	9
2.1.6	Skewness.....	10
2.2	Related work.....	10
2.2.1	Studies relating to cryptocurrency price prediction with deep learning.....	10
2.2.2	Studies relating to cryptocurrency candlestick price prediction with deep learning	13
2.2.3	Normalization technique in cryptocurrency price prediction with deep learning.....	13
2.2.4	Studies relating to candlestick price and candlestick price prediction method.....	14
Chapter 3:	Methodology.....	17
3.1	Problem approach.....	17
3.1.1	Data source.....	18
3.1.2	Data preparation.....	18
3.1.3	Feature construction.....	19
3.1.4	Model construction .....	19
3.1.5	Tools and programming language.....	21
3.2	Model performance improvement with normalization techniques.....	21
3.3	Candlestick price prediction method .....	23
Chapter 4:	Result and discussion.....	26



4.1	Exploratory data analysis .....	26
4.1.1	Exploratory data analysis before data splitting.....	26
4.1.2	Exploratory data analysis after data splitting .....	27
4.2	Model performance.....	28
4.3	Model performance improvement with various normalization techniques .....	29
4.4	Impact of normalization on data characteristics .....	32
4.4.1	Impact of normalization on data distribution .....	32
4.4.2	Impact of normalization on sliding window .....	35
4.5	Exploring candlestick price prediction method .....	42
4.5.1	Exploring the model with CULR feature .....	42
4.5.1.1	Exploratory data analysis for CULR .....	42
4.5.1.2	Model performance with CULR feature .....	47
4.5.2	Forecasting error comparison.....	50
4.5.3	Accuracy comparison .....	51
4.5.4	The best model performance.....	52
4.6	Exploring effect of sliding window size.....	54
4.7	Performance comparison of proposed method with previous research models.....	56
4.8	Exploring model performance with before the cryptocurrency boom (before Aug 18, 2020)	
	58	
4.8.1	Exploratory data analysis for data before Aug 18, 2020.....	58
4.8.2	Model performance with various normalization techniques.....	60
4.8.3	Impact of normalization on data characteristics .....	62
4.8.3.1	Impact of normalization on data distribution.....	62
4.8.3.2	Impact of normalization on sliding window.....	66
4.8.3.3	Summary of before the cryptocurrency boom .....	72
Chapter 5:	Conclusion .....	73
Appendix.....		74

REFERENCES ..... 77

VITA..... 80

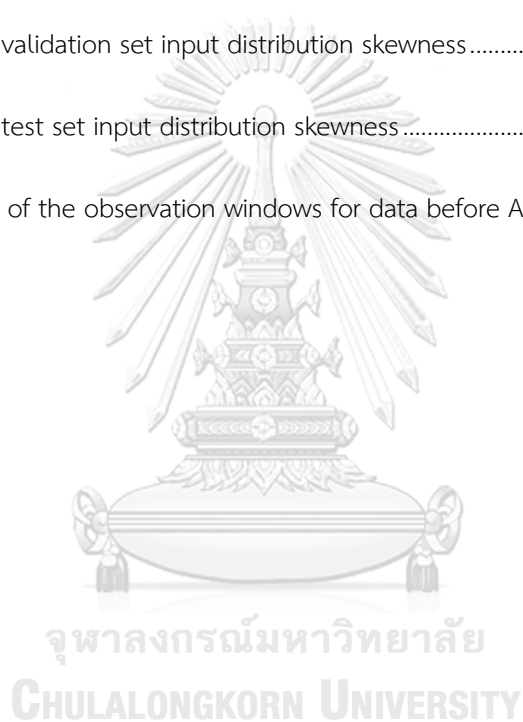


## LIST OF TABLES

	Page
Table 1 Confusion matrix.....	8
Table 2 Pearson's second coefficient of skewness value interpretation.....	10
Table 3 Studies relating to cryptocurrency price prediction with deep learning.....	11
Table 4 Studies relating to cryptocurrency candlestick price prediction with deep learning.....	13
Table 5 Studies relating to cryptocurrency candlestick price prediction with deep learning.....	13
Table 6 Studies relating to candlestick price and candlestick price prediction method.....	15
Table 7 Raw data definitions.....	18
Table 8 The tuned hyperparameter and tuning range.....	21
Table 9 The fixed hyperparameter.....	21
Table 10 Basic statistical summary table of each price.....	27
Table 11 Basic statistical summary table of open price after data splitting.....	27
Table 12 Average and standard deviation of MAPE (%) on test set for LSTM and GRU.....	29
Table 13 Average and standard deviation of RMSE on test set for LSTM and GRU.....	29
Table 14 LSTM average and standard deviation of MAPE (%) on the test set.....	30
Table 15 LSTM average and standard deviation of RMSE on the test set.....	30
Table 16 GRU average and standard deviation of MAPE (%) on the test set.....	30
Table 17 GRU average and standard deviation of RMSE the test set.....	31
Table 18 Average of training set input distribution skewness.....	33
Table 19 Average of validation set input distribution skewness.....	33

Table 20 Average of test set input distribution skewness .....	33
Table 21 Time range of the observation windows.....	38
Table 22 Summary statistics of each feature .....	42
Table 23 Summary statistics for close price after data splitting .....	44
Table 24 Summary statistics for upper shadow after data splitting.....	44
Table 25 Summary statistics for lower shadow after data splitting .....	45
Table 26 Summary statistics for real body after data splitting.....	45
Table 27 LSTM average and standard deviation of MAPE on the test set .....	48
Table 28 LSTM average and standard deviation of RMSE on the test set.....	48
Table 29 GRU average and standard deviation of MAPE on the test set.....	48
Table 30 GRU average and standard deviation of RMSE on the test set.....	49
Table 31 Average and standard deviation of accuracy on the test set.....	52
Table 32 RMSE and MAPE of best test set RMSE.....	53
Table 33 Classification metrics of best test set RMSE .....	53
Table 34 RMSE and MAPE of best test set Accuracy .....	53
Table 35 Classification metrics of best test set Accuracy.....	53
Table 36 Summary of the sliding window effect on the forecasting performance characteristic..	56
Table 37 A comparison of our purposed model with the previous research model.....	57
Table 38 Summary statistics of each price.....	59
Table 39 Summary statistics for open price before Aug 18, 2020 after data splitting .....	59
Table 40 LSTM average and standard deviation of MAPE (%) on the test set using data before Aug 18, 2020. ....	60

Table 41 LSTM average and standard deviation of RMSE on the test set using data before Aug 18, 2020.....	60
Table 42 GRU average and standard deviation of MAPE (%) on the test set using data before Aug 18, 2020.....	61
Table 43 GRU average and standard deviation of RMSE the test set before data before Aug 18, 2020.....	61
Table 44 Average of training set input distribution skewness.....	63
Table 45 Average of validation set input distribution skewness.....	63
Table 46 Average of test set input distribution skewness.....	64
Table 47 Time range of the observation windows for data before Aug 18, 2020 .....	69



## LIST OF FIGURES

	Page
Figure 1 A simple RNN unit representation (Afshine Amidi) .....	3
Figure 2 An internal Long short-term memory (LSTM) representation (Afshine Amidi).....	4
Figure 3 An internal Gated Recurrent Unit (GRU) representation(Afshine Amidi) .....	6
Figure 4 Workflow.....	17
Figure 5 Feature diagram of the model.....	20
Figure 6 Structure diagram of the model (Afshine Amidi).....	20
Figure 7 Workflow of each normalization approach .....	22
Figure 8 Candlestick component (16 must-know candlestick patterns for a successful trade, 2021) .....	24
Figure 9 Workflow of CULR method .....	24
Figure 10 Time series plot of OHLC prices .....	26
Figure 11 The decomposition plot of open price.....	27
Figure 12 Time series plot of open prices with data splitting.....	28
Figure 13 Distribution of open price with data splitting .....	28
Figure 14 Box plot showing average RMSE of OHLC with various normalization techniques in GRU and LSTM.....	31
Figure 15 Box plot showing each price RMSE with various normalization techniques in GRU and LSTM.....	31
Figure 16 Sample input distribution before normalization of (a) training set (b) validation set (c) test set .....	32

Figure 17 Skewness of training set open price input distribution with various normalization techniques.....	33
Figure 18 Skewness of validation set open price input distribution with various normalization techniques.....	34
Figure 19 Skewness of test set open price input distribution with various normalization techniques.....	34
Figure 20 Sample input distribution after normalization by (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change.....	35
Figure 21 Sliding window mean after normalization by (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change.....	36
Figure 22 Sliding window standard deviation after normalization by (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change.....	37
Figure 23 Time series plot of open price with selected sliding windows for observation.....	38
Figure 24 Time series plot of open price sliding window without normalization at each observation window.....	39
Figure 25 Time series plot of open price sliding window with z-score normalization techniques at each observation window.....	39
Figure 26 Time series plot of open price sliding window with min-max normalization techniques at each observation window.....	40
Figure 27 Time series plot of open price sliding window with relative change normalization techniques at each observation window.....	40
Figure 28 Box plot of open price sliding window with various normalization at each observation window a) without normalization b) z-score normalization c) min-max normalization d) relative change normalization.....	41

Figure 29 Time series plot of CULR (a) close price (b) upper shadow (c) lower shadow (d) real body..... 42

Figure 30 The decomposition plot of close price ..... 43

Figure 31 The decomposition plot of upper shadow ..... 43

Figure 32 The decomposition plot of lower shadow..... 43

Figure 33 The decomposition plot of upper shadow ..... 44

Figure 34 Visualization of close price with data splitting (a) Time series plot (b) Histogram of training set (c) Histogram of validation set (d) Histogram of test set ..... 45

Figure 35 Visualization of upper shadow with data splitting (a) time series plot (b) histogram of training set (c) histogram of validation set (d) histogram of test set ..... 46

Figure 36 Visualization of lower shadow with data splitting (a) time series plot (b) histogram of training set (c) histogram of validation set (d) histogram of test set ..... 46

Figure 37 Visualization of real body with data splitting (a) time series plot (b) histogram of training set (c) histogram of validation set (d) histogram of test set ..... 47

Figure 38 Box plot showing RMSE of each price with various normalization techniques on CULR Method..... 49

Figure 39 Box plot showing average RMSE of OHLC price with various normalization techniques on CULR Method ..... 49

Figure 40 Box plot showing each price RMSE with various normalization techniques for candlestick prediction method performance comparison..... 50

Figure 41 Box plot showing average RMSE of OHLC with various normalization techniques for candlestick price prediction method performance comparison. .... 51

Figure 42 Box plot of test set accuracy with various candlestick prediction method ..... 52

Figure 43 Visualization of the effect of sliding window size on RMSE with various normalization techniques for the OHLC method (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change ..... 55



Figure 44 Visualization of the effect of sliding window size on RMSE with various normalization techniques for the CULR method (a) whole set z-score (b) whole set min-max (c) window z-score (d) window min-max .....	55
Figure 45 Time series plot of OHLC prices before Aug 18, 2020.....	58
Figure 46 Visualization of open price before Aug 18, 2020 with data splitting (a) time series plot (b) histogram of training set (c) histogram of validation set (d) histogram of test set.....	59
Figure 47 Box plot showing RMSE of each price with various normalization techniques in GRU and LSTM models using data before Aug 18, 2020 .....	62
Figure 48 Box plot showing average RMSE of OHLC with various normalization techniques in GRU and LSTM models using data before Aug 18, 2020.....	62
Figure 49 Sample input distribution before normalization of (a) training set (b) validation set (c) test set .....	63
Figure 50 Skewness of training set open price input distribution with various normalization techniques.....	64
Figure 51 Skewness of validation set open price input distribution with various normalization techniques.....	64
Figure 52 Skewness of test set open price input distribution with various normalization techniques.....	65
Figure 53 Sample input distribution after normalization by (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change.....	66
Figure 54 Sliding window mean after normalization by (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change.....	67
Figure 55 Sliding window standard deviation after normalization by (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change .....	67

Figure 56 Time series plot of open price with selected sliding windows for observation.....	68
Figure 57 Time series plot of open price sliding window without normalization at each observation window.....	69
Figure 58 Time series plot of open price sliding window with z-score normalization techniques at each observation window.....	70
Figure 59 Time series plot of open price sliding window with min-max normalization techniques at each observation window.....	70
Figure 60 Time series plot of open price sliding window with relative change normalization techniques at each observation window.....	71
Figure 61 Box plot of open price sliding window with various normalization at each observation window a) without normalization b) z-score normalization c) min-max normalization d) relative change normalization .....	71



# Chapter 1: Introduction

## 1.1 Background

As a financial asset, cryptocurrencies have higher volatility than other assets, especially in recent years (Ghorbel & Jeribi, 2021). Cryptocurrencies are digital or virtual currencies that are secured by cryptography and designed to be the medium of exchange as a fiat currency. Investing in cryptocurrencies can help diversify portfolio risks. Furthermore, most cryptocurrencies generally provide better average daily returns than traditional investments (Kuo Chuen et al., 2017).

One of the most well-known cryptocurrencies is Bitcoin. It is the world's first cryptocurrency and was introduced in October 2008 by Satoshi Nakamoto (Nakamoto & Bitcoin, 2008). Moreover, Bitcoin is the world's largest market-cap cryptocurrency and the highest price coin. Bitcoin trading is a viable option for investment due to its popularity and credibility. In order to aid a trader's decision in buying or selling an investment instrument, forecasting an asset price movements is a common practice (Alkhodhairi et al., 2021).

Candlestick price or OHLC price is a type of price representation that displays four prices for a specific time interval. The candlestick price contains the opening price, the highest price, the lowest price, and the closing price. The candlestick price prediction assists traders in making more sophisticated decisions by buying financial assets at near-predicted low prices and selling at near-predicted high prices (Wang et al., 2021). In addition, the candlestick pattern can reflect the market circumstances and sentiment.

Data normalization is a pre-processing step that transforms data into a specific range and, in some cases, also modifies data distribution. Data normalization is a crucial process in neural networks and other machine learning algorithms to ensure the quality of data before it is given to a model (Panigrahi & Behera, 2013). Normalization can also help reduce the training time, as it starts the training process with each feature on the same scale, as well as reduce bias inside the neural network from one feature to another (Nayak et al., 2013). Research shows that model performance may vary when different normalization techniques are used on the same model setting (Nayak et al., 2013; Panigrahi & Behera, 2013).

From the benefit of Bitcoin trading and owning the Bitcoin, finding the right time to trade can help the trader get more profit. A forecast is a tool for decision making, so knowing more information like candlestick price helps the trader make a better decision. In addition, the higher quality of data given to a model can improve model forecasting performance.

## 1.2 Objectives of the study

1. To create a model to predict the Bitcoin candlestick price for the next period using the neural network technique.
2. To explore feature transformation aiming to enhance the forecasting performance of the neural network. Specifically, various data normalization (scaling) such as z-score, min-max, and relative change normalization, will be investigated.
3. To explore additional candlestick price prediction approaches that are not OHLC price prediction, for example, candle wick (CULR) prediction, etc.

## 1.3 Scope of the study

1. This study will analyze Bitcoin daily price during August 2017 – August 2022 on the Binance trade exchange.
2. This study will focus on quantitative methodologies and will investigate only endogenous variables.
3. The investigated neural network technique is a recurrent neural network, which includes Long short-term memory (LSTM) and Gated recurrent unit (GRU).
4. The model performance improvement method in this study is the feature transformation technique, and the investigated feature transformation technique is data normalization (scaling), such as Z-Score, Min-Max, and relative change normalization.
5. The forecasting performance metrics in this study are Mean absolute percentage error (MAPE) and Root mean square error (RMSE).

## 1.4 Expected outcomes of the study

1. Obtain a model to predict the Bitcoin candlestick price
2. Understand the effectiveness of various normalization techniques
3. Acquire different approaches to create candlestick price prediction

## 1.5 Expected benefits of the study

1. Accurate candlestick price forecasting helps the investor to make a higher rate of return.
2. The normalization technique developed in this research can be applied to other time series data using neural networks in the future.
3. The candlestick price prediction method developed in this research can help the investor to make a better decision.

## Chapter 2: Related theories and literature review

### 2.1 Related theories

#### 2.1.1 Neural network

The neural network is a type of algorithm in artificial intelligence that performs data processing in a manner inspired by the human brain. It is a subset of machine learning and the foundation of deep learning algorithms.

#### 2.1.2 Recurrent neural network (RNN)

Recurrent neural networks are neural networks that have been modified to handle sequence data. It used the previous state (memory) as a feature of the current state, so its computation considers historical information.

##### 2.1.2.1 Simple RNN

A simple recurrent neural network or Simple RNN is a basic type of RNN that receives input data ( $x^{<t>}$ ) and the previous hidden state ( $a^{<t-1>}$ ) before passing them to the activation function ( $g_1$ ), which is used to update the hidden state. The hidden state can be calculated by equation 2.1 and the output by equation 2.2. A pictorial representation of a single simple RNN unit is shown in Figure 1.

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad (2.1)$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y) \quad (2.2)$$

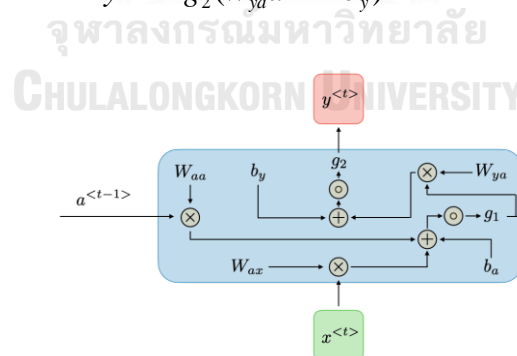


Figure 1 A simple RNN unit representation (Afshine Amidi)

Where  $a^{<t>}$  = current cell hidden state

$x^{<t>}$  = current cell input data

$W_{aa}$ ,  $W_{ax}$ ,  $W_{ya}$  are weighted of hidden layers

$b_a$ ,  $b_y$  are bias of hidden layers

$g_1$  = activation function for hidden state

$g_2$  = activation function for output

### 2.1.2.2 Long short-term memory (LSTM)

Long short-term memory is a type of RNN which is capable of handling long-term dependencies. The core principles of LSTM are gate mechanisms and cell states. The cell state is a unit memory of the network. The gates are mathematical functions that control information flow in the cell and the information memorizing process. There are three types of gates in LSTM: input gate, output gate, and forget gate. A pictorial representation of a single Long Short-Term Memory unit is shown in Figure 2.

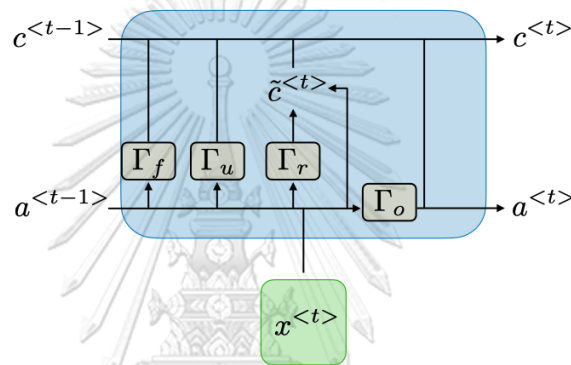


Figure 2 An internal Long short-term memory (LSTM) representation (Afshine Amidi)

The gates and cell states are composed of mathematical functions and operations that serve as the basis for data manipulation. The related functions are the hyperbolic tangent ( $\tanh$ ) and the sigmoid function ( $\sigma$ ), which are non-linear functions. Tanh regulates the network's values, keeping them between -1 and 1, whereas sigmoid function output values range from 0 to 1.

The Hadamard product, also known as the element-wise product ( $*$ ), is an important operator in this algorithm. It is an operation that takes two matrices of the same dimensions and multiplies each element corresponding to the same row and columns.

The cell state or memory cell collects information from the previous cell state ( $c^{<t-1>}$ ) and passes current cell state ( $c^{<t>}$ ) to the next cell. The current cell state ( $c^{<t>}$ ) is calculated by combining a candidate cell state ( $\tilde{c}^{<t>}$ ) which is considered as new information and the previous cell state ( $c^{<t-1>}$ ). The candidate cell state and the current cell state can be calculated by equations 2.3 and 2.7 respectively.

The forget gate ( $\Gamma_f$ ) determines whether the cell should retain the previous cell state ( $c^{<t-1>}$ ) or discard it via looking at input from the current timestep ( $x^{<t>}$ ) and the old hidden state ( $a^{<t-1>}$ ). Since the activation function of the forget gate is the sigmoid function, the output

value of this gate is a number between 0 and 1. When the output value is close to 0, the cell forgets the previous cell state ( $c^{<t-1>}$ ). When the value is 1, it indicates that the cell must store data. The forget gate is shown in equation 2.5.

The input gate or update gate ( $\Gamma_u$ ) determines how much to update the previous cell state with a new candidate ( $\tilde{c}^{<t>}$ ) by watching the inputs from the current timestep ( $x^{<t>}$ ) and the old hidden state ( $a^{<t-1>}$ ) like the forget gate. In addition, the update gate works together with the forget gate to create the current cell state to pass the value to the next cell. The input gate can be calculated by equation 2.4.

The output gate ( $\Gamma_o$ ), which can be calculated by equation 2.6, regulates how much to reveal the current cell state ( $c^{<t>}$ ) to the current cell hidden state ( $a^{<t>}$ ). Like the other gates, the output gate value can be calculated by the current timestep ( $x^{<t>}$ ) and the old hidden state ( $a^{<t-1>}$ ). Then the current hidden state ( $a^{<t>}$ ) is generated by the output gate and tanh function of the current cell state as shown in equation 2.8.

$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c) \quad (2.3)$$

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u) \quad (2.4)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f) \quad (2.5)$$

$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o) \quad (2.6)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>} \quad (2.7)$$

$$a^{<t>} = \Gamma_o * \tanh(c^{<t>}) \quad (2.8)$$

Where  $\tilde{c}^{<t>}$  = candidate cell state,  $c^{<t>}$  = current cell state,

$a^{<t>}$  = current hidden state,  $x^{<t>}$  = the current timestep input

$\Gamma_u$  = update gate,  $\Gamma_f$  = forgot gate,  $\Gamma_o$  = output gate

$W_c$  = cell state weight,  $W_u$  = update gate weight,  $W_f$  = forgot gate weight,

$W_o$  = output gate weight,  $b_c$  = cell state bias,  $b_u$  = update gate bias,

$b_f$  = forgot gate bias, and  $b_o$  output gate bias

### 2.1.2.3 Gated recurrent unit (GRU)

Gated recurrent unit (GRU) uses the same core principle as the LSTM but has a different architecture and fewer parameters. GRU has only two types of gates: update gate and relevance gate. A pictorial representation of a single Gated recurrent unit is shown in Figure 3.

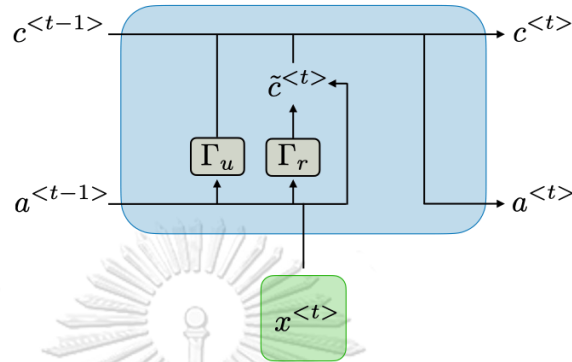


Figure 3 An internal Gated Recurrent Unit (GRU) representation(Afshine Amidi)

The relevance gate ( $\Gamma_r$ ), also known as the reset gate, determines the importance of the previous cell state ( $c^{<t-1>}$ ) in candidate cell state ( $\tilde{c}^{<t>}$ ) calculation. The prior timestep cell state ( $c^{<t-1>}$ ) and the present input data ( $x^{<t>}$ ) are parameters of this gate. The relevance gate is represented by equation 2.11, while the candidate cell is represented by equation 2.9.

The update gate ( $\Gamma_u$ ), which can be calculated by equation 2.10, decides how much to update a new candidate ( $\tilde{c}^{<t>}$ ) with the previous cell state by watching the inputs from the current timestep ( $x^{<t>}$ ) and the previous cell state ( $c^{<t-1>}$ ). Inversely, the update gate determines the amount to discard the prior timestep cell state as demonstrated in equation 2.12. Unlike LSTM, the current hidden cell state in GRU is equal to the present cell state, as shown in equation 2.13.

$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c) \quad (2.9)$$

$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u) \quad (2.10)$$

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r) \quad (2.11)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>} \quad (2.12)$$

$$a^{<t>} = c^{<t>} \quad (2.13)$$

Where  $\tilde{c}^{<t>}$  = candidate cell state,  $c^{<t>}$  = current cell state,

$a^{<t>}$  = current hidden state,  $x^{<t>}$  = the current timestep input



$\Gamma_u$  = update gate,  $\Gamma_f$  = forgot gate,  $\Gamma_r$  = relevant gate

$W_c$  = cell state weight,  $W_u$  = update gate weight,  $W_r$  = relevant gate weight,

$b_c$  = cell state bias,  $b_u$  = update gate bias,  $b_r$  = relevant gate bias

### 2.1.3 Forecasting performance metric

When evaluating the performance of a time series model, there is a mathematical notation is developed for forecasting which is summarized as follows:

$Y_t$  = the actual value of a time series at period t

$\hat{Y}_t$  = the forecast value for period t

$e_t = Y_t - \hat{Y}_t$  = the residual or forecast error

#### 2.1.3.1 Mean absolute error (MAE)

The mean absolute error (MAE) can be calculated by averaging the absolute error as shown in equation 2.14. It provides an average size of the miss without direction consideration and measures the accuracy by averaging the magnitudes of the error.

$$MAE = \frac{1}{n} \sum_{t=1}^n |Y_t - \hat{Y}_t| \quad (2.14)$$

#### 2.1.3.2 Mean absolute percentage error (MAPE)

The mean absolute percentage error (MAPE) is calculated by averaging the ratio of the absolute error in each period and the absolute of the actual value, as shown in equation 2.15. The final value is then multiplied by 100 and expressed as a percentage.

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|Y_t - \hat{Y}_t|}{Y_t} \times 100\%, \quad Y_t \neq 0 \quad (2.15)$$

#### 2.1.3.3 Mean square error (MSE)

The mean square error (MSE) is determined by averaging the squared of error, as given in equation 2.16. This measurement penalizes large forecasting errors.

$$MSE = \frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}_t)^2 \quad (2.16)$$

### 2.1.3.4 Root mean square error (RMSE)

The root mean square error (RMSE) is determined by square root of averaging the squared of error, as given in equation 2.17. This measurement penalizes large forecasting errors. This error has the same unit as the predictor and inherits the MSE's characteristic in penalizing large forecasting errors.

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}_t)^2} \quad (2.17)$$

### 2.1.4 Classification metrics

The classification metrics are used to evaluate the performance of classification problems and are based on the confusion matrix. The confusion matrix is a table that shows the number of true positives, true negatives, false positives, and false negatives predictions made by the model compared to the actual labels, as shown in Table 1.

Table 1 Confusion matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

#### 2.1.4.1 Accuracy

Accuracy is the proportion of correctly classified samples out of all samples, as shown in equation 2.18.

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \quad (2.18)$$

#### 2.1.4.2 Precision

Precision is the proportion of true positives out of all predicted positives, as given in equation 2.19.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.19)$$

#### 2.1.4.3 Recall

Recall is the proportion of true positives out of all actual positives, as given in equation 2.20.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.20)$$

#### 2.1.4.4 F1-Score

F1 score is the harmonic mean of precision and recall, balances both metrics, as shown in equation 2.21 – 2.22.

$$\text{F1} = 2 \times \left( \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \quad (2.21)$$

$$\text{F1} = \frac{2TP}{2TP + FP + FN} \quad (2.22)$$

### 2.1.5 Data normalization and scaling techniques

#### 2.1.5.1 Z-Score normalization

To normalize the data, this technique uses the mean ( $\mu$ ) and standard deviation ( $\sigma$ ) of the original data as shown in equation 2.23. The training set's mean and standard deviation were utilized for z-score normalization.

$$x' = \frac{x - \mu}{\sigma} \quad (2.23)$$

#### 2.1.5.2 Min-Max normalization

As demonstrated in equation 2.24, the minimum value ( $x_{\min}$ ) and maximum value ( $x_{\max}$ ) of the original data are used to linear transform the data. The training set's minimum and maximum are utilized to transform the training set and the other sets for the min-max normalization.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (2.24)$$

#### 2.1.5.3 Relative change normalization

The relative change normalizes the window elements by using the first value of the sequence ( $x_0$ ) as illustrated in equation 2.25. In general case, the first value of the sequence is the first value of the training set.

$$x' = \frac{x - x_0}{x_0} \quad (2.25)$$

### 2.1.6 Skewness

Skewness is a statistical property that measures the asymmetry of the probability distribution. This study utilizes Pearson's second coefficient of skewness to measure the skewness which can be calculated by equation 2.26. The skewness value interpretation is shown in Table 2.

$$skewness = \frac{3(\bar{x} - median)}{s} \quad (2.26)$$

Table 2 Pearson's second coefficient of skewness value interpretation

Level of skewness	Skewed left (negatively skewed)	Skewed right (positively skewed)
Approximately symmetric	$-0.5 < skewness < 0$	$0 < skewness < 0.5$
Moderately skewed	$-1.0 < skewness < -0.5$	$0.5 < skewness < 1.0$
Highly skewed	$skewness < -1.0$	$skewness > 1.0$

## 2.2 Related work

### 2.2.1 Studies relating to cryptocurrency price prediction with deep learning

Many researchers have explored various techniques and algorithms to predict the cryptocurrency market. Some of which involve neural networks. GRU, LSTM, and MLP were used to predict the Bitcoin's next-hour price by using 24 hours of data as input (Jiang, 2020). They were compared by changing the neural network cell type and the number of layers. The results revealed that the LSTM 2 layers outperform all others, and GRU 2 layers outperform all others when tested on cross validation. Another study examined the performance of LSTM and GRU by altering the size of the window and the number of days ahead in a day trading period. The study concluded that the optimal settings for this experiment were a window size of 12 and days ahead of 7 performed by LSTM, while GRU outperformed LSTM in other situations (Muniye et al., 2020)

Some papers are talking about a comparison of traditional forecasting techniques and neural network techniques. The classical forecasting method, linear regression, was compared to the Long Short-Term Memory (LSTM) in (Kavitha et al., 2020), and it was discovered that the LSTM outperformed the old technique in terms of R2, MAE, and RMSE. In (Phaladisailoed & Numnonda, 2018), the Gated Recurrent Unit, and regression methods: Theil-Sen Regression and Huber Regression, were compared to LSTM. The recurrent neural networks outperformed the regressions in terms of R2, and GRU had the best MSE while MSE of the regressions were better than the LSTM. A well-known time series forecasting model such as ARIMA was also compared to

modern techniques such as simple recurrent neural networks (RNN) and LSTM in terms of RMSE and classification performance, where the labels for categorization were price up, price down, and no change. Z-score normalization technique was applied. Deep learning algorithms outperformed ARIMA in terms of RMSE and accuracy, but not precision (McNally et al., 2018). Five cryptocurrencies' close prices, including Bitcoin price, were predicted by OHLC prices and volume (Hansun et al., 2022). They also compared the performance of LSTM, GRU, and bidirectional LSTM (Bi-LSTM) using the min-max normalization. The results show that GRU outperformed the others with an RMSE of 1,777.31.

A novel hybrid technique based on LSTM and GRU for cryptocurrency prediction was applied to predict Litecoin and Zcash's next-day price by using the previous days' weighted price (mean of OHLC price) and the Bitcoin price direction (Tanwar et al., 2021). This study also tested the algorithm with various window sizes (1, 3, 7, and 30 days) and compared the novel technique to Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM). The results showed that the novel technique outperformed the GRU and LSTM in both Litecoin and Zcash prediction.

There is also a study that examined the effect of an external factor in cryptocurrency prediction with the recurrent neural networks, which are simple RNN, LSTM, and GRU (Vanderbilt et al., 2020). They compared using the previous-day price to predict the next-day price with using the previous-day price and google trends data and tested on three cryptocurrencies: Bitcoin, Ripple, and Litecoin. The study concluded that Google trends data is not a useful data input for RNN models in cryptocurrency price prediction, and GRU outperforms all others. The summary of this topic is shown in Table 3.

*Table 3 Studies relating to cryptocurrency price prediction with deep learning*

Authors	Data	Features	Output	Techniques	Results
(Jiang, 2020)	Bitcoin minute price but convert to hourly price (Jan 2012 to July 2018)	Previous 24-hour price	Next hour price	Multi-Layer Perceptron (MLP), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM)	- The best RMSE: <b>19.020</b> by LSTM
(Muniye et al., 2020)	Bitcoin daily price	Previous day price in different window size	Next day price in different number of days ahead	Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM)	- The best RMSE: <b>0.045</b> - The best MAPE: <b>0.030</b> by LSTM
(Kavitha et al., 2020)	Bitcoin minute price but convert to a day period price (Jun 2012 to July 2019)	Previous day weighted price (Mean of OHLC price)	Next day weighted price (Mean of OHLC price)	Long Short-Term Memory (LSTM), Linear Regression	- LSTM: RMSE = 95.067, MAE = 64.389, $R^2 = 0.981$ - Linear Regression: RMSE = 296.747, MAE

Authors	Data	Features	Output	Techniques	Results
					=143.704, $R^2 = 0.908$
(Phaladisailoed & Numnonda, 2018)	Bitcoin minute price but convert to a day period price (Jan 2012 to Jan 2018)	OHLC price	Next day weighted price (Mean of OHLC price)	Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Theil-Sen Regression, Huber Regression	<ul style="list-style-type: none"> <li>- <b>LSTM:</b> MSE = 0.000431, <math>R^2 = 0.992</math></li> <li>- <b>GRU:</b> MSE = 0.00002, <math>R^2 = 0.992</math></li> <li>- <b>Theil-Sen Regression:</b> MSE = 0.000375, <math>R^2 = 0.99176</math></li> <li>- <b>Huber Regression:</b> MSE = 0.000373, <math>R^2 = 0.99179</math></li> </ul>
(McNally et al., 2018)	Bitcoin daily close price (August 2013 to July 2016)	<ul style="list-style-type: none"> <li>- SMA 5 days close price</li> <li>- SMA 10 days close price</li> <li>- de-noised close price</li> </ul>	Next day close price	Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Autoregressive integrated moving average (ARIMA)	<ul style="list-style-type: none"> <li>- <b>LSTM:</b> RMSE = 6.87%</li> <li>Precision = 35.50%</li> <li>- <b>RNN:</b> RMSE = 5.45%</li> <li>Precision = 39.08%</li> <li>- <b>ARIMA:</b> RMSE = 53.74%</li> <li>Precision = 100 %</li> </ul>
(Hansun et al., 2022)	Bitcoin daily price (Sept 2014 to Oct 2021)	- OHLC price and volume	Next day close price	<ul style="list-style-type: none"> <li>- Gated Recurrent Unit (GRU)</li> <li>- Long Short-Term Memory (LSTM)</li> <li>- Bidirectional Long Short-Term Memory (Bi-LSTM)</li> </ul>	<ul style="list-style-type: none"> <li>- <b>LSTM:</b> RMSE=2518.02</li> <li>MAPE = 4.23%</li> <li>- <b>Bi-LSTM:</b> RMSE = 2222.73</li> <li>MAPE = 3.80%</li> <li>- <b>GRU:</b> RMSE = 1777.30</li> <li>MAPE = 3.50%</li> </ul>
(Tanwar et al., 2021)	Daily price of - Litecoin (Aug 2016 - May 2021) - Zcash (Oct 2016 - May 2021) - Bitcoin	<ul style="list-style-type: none"> <li>- weighted cryptocurrency price and parent coin's direction (Bitcoin direction: up and down)</li> <li>** tested algorithm with various window size: 1,3,7, and 30 days</li> </ul>	Next day cryptocurrency price	<ul style="list-style-type: none"> <li>- Gated Recurrent Unit (GRU)</li> <li>- Long Short-Term Memory (LSTM)</li> <li>- LSTM-GRU hybrid</li> </ul>	<ul style="list-style-type: none"> <li><b>Litecoin:</b></li> <li>- <b>GRU:</b> MSE = 0.02113 (1 day)</li> <li>- <b>LSTM:</b> MSE = 0.0285 (1 day)</li> <li>- <b>Hybrid:</b> MSE = 0.02038 (1 day)</li> <li><b>Zcash:</b></li> <li>- <b>GRU:</b> MSE = 0.00462 (7 days)</li> <li>- <b>LSTM:</b> MSE = 0.00497 (3 days)</li> <li>- <b>Hybrid:</b> MSE = 0.00461 (1 day)</li> </ul>
(Vanderbilt et al., 2020)	Daily price of - Litecoin - Ripple - Bitcoin (Jan 2015 - Apr 2020)	Compare two methods - previous day price - previous day price with Google trends data	Next day price	<ul style="list-style-type: none"> <li>- Simple RNN</li> <li>- Gated Recurrent Unit (GRU)</li> <li>- Long Short-Term Memory (LSTM)</li> </ul>	<ul style="list-style-type: none"> <li><b>Bitcoin:</b></li> <li>- <b>RNN:</b> RMSE = 569.02</li> <li>- <b>RNN with google trend:</b> RMSE = 569.41</li> <li>- <b>LSTM:</b> RMSE = 562.08</li> <li>- <b>LSTM with google trend:</b> RMSE = 553.19</li> </ul>

Authors	Data	Features	Output	Techniques	Results
					- GRU: RMSE = 403.23 - GRU with google trend: RMSE = 411.09

## 2.2.2 Studies relating to cryptocurrency candlestick price prediction with deep learning

Limited research has been on candlestick or OHLC Bitcoin price prediction. While most studies on the candlestick price utilized it as a feature to predict the price, (Alkhodhairi et al., 2021) employed the candlestick as both a feature and an output. The study examined the performance of LSTM and GRU in predicting OHLC price, as well as determined the candlestick interval (4 h, 12 h, or 24 h) that rendered the most accurate forecast. This study used decimal scaling for feature normalization and used a novel technique called real-time data prediction, which involves feeding new data to the whole dataset and fitting the data again to update the model. The findings showed that the LSTM with a 4-h interval along with the real-time data prediction technique yielded an optimal performance, as shown in Table 4.

*Table 4 Studies relating to cryptocurrency candlestick price prediction with deep learning*

Authors	Data	Features	Output	Techniques	Results
(Alkhodhairi et al., 2021)	Bitcoin minute price but convert to 4h, 12h, 24h price (Jan 2017 to Aug 2020)	Previous interval OHLC	Next interval OHLC	Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU)	- Real time prediction MAPE: <b>0.63 % by LSTM</b> - Historical prediction MAPE: <b>2.60 % by LSTM</b>

## 2.2.3 Normalization technique in cryptocurrency price prediction with deep learning

Table 5 shows an overview of previous research's normalization techniques according to 2.2.1 - 2.2.2. In the investigated studies, four techniques appear: Min-Max normalization, Z-score normalization, relative change normalization, and Decimal scaling. In addition, the most popular normalization technique in the explored studies is Min-Max scaling.

*Table 5 Studies relating to cryptocurrency candlestick price prediction with deep learning*

Authors	Data	Algorithm	Technique
(Jiang, 2020)	Bitcoin minute price but convert to hourly price (Jan 2012 to July 2018)	- Multi-Layer Perceptron (MLP) - Gated Recurrent Unit (GRU) - Long Short-Term Memory (LSTM)	- Min-Max - Relative change
(Muniye et al., 2020)	Bitcoin daily price (Jan 2014 to Feb 2018)	- Gated Recurrent Unit (GRU) - Long Short-Term Memory (LSTM)	Min-Max

Authors	Data	Algorithm	Technique
(Phaladisailoed & Numnonda, 2018)	<b>Bitcoin</b> minute price but convert to a day period price (Jun 2012 to July 2019)	- Long Short-Term Memory (LSTM) - Linear Regression	Min-Max
(Kavitha et al., 2020)	<b>Bitcoin</b> minute price but convert to a day period price (Jan 2012 to Jan 2018)	- Gated Recurrent Unit (GRU) - Long Short-Term Memory (LSTM), - Theil-Sen Regression - Huber Regression	Min-Max
(McNally et al., 2018)	<b>Bitcoin</b> daily close price (August 2013 to July 2016)	- Recurrent Neural Network (RNN) - Long Short-Term Memory (LSTM) - Autoregressive integrated moving average (ARIMA)	Z-Score
(Hansun et al., 2022)	<b>Bitcoin</b> daily price (Sept 2014 to Oct 2021)	- Gated Recurrent Unit (GRU) - Long Short-Term Memory (LSTM) - Bidirectional Long Short-Term Memory (Bi-LSTM)	Min-Max
(Tanwar et al., 2021)	Daily price of - <b>Litecoin</b> (Aug 2016 - May 2021) - <b>Zcash</b> (Oct 2016 - May 2021) - <b>Bitcoin</b> (Aug 2016 - May 2021)	- Gated Recurrent Unit (GRU) - Long Short-Term Memory (LSTM) - LSTM-GRU hybrid	Z-Score
(Vanderbilt et al., 2020)	Daily price of - <b>Litecoin</b> (Jan 2015 - Apr 2020) - <b>Ripple</b> (Jan 2015 - Apr 2020) - <b>Bitcoin</b> (Jan 2015 - Apr 2020)	- Simple RNN - Gated Recurrent Unit (GRU) - Long Short-Term Memory (LSTM)	Min-Max
(Alkhodhairi et al., 2021)	<b>Bitcoin</b> minute price but convert to 4h, 12h, 24h price (Jan 2017 to Aug 2020)	- Long Short-Term Memory (LSTM) - Gated Recurrent Unit (GRU)	Decimal scaling

#### 2.2.4 Studies relating to candlestick price and candlestick price prediction method

Some previous works explored candlestick price prediction methods. (Alkhodhairi et al., 2021) used the previous time interval OHLC price to predict the next OHLC price with LSTM and GRU on Bitcoin data. They used MAPE, RMSE, MAE, and R2 to measure the performance of the model. Another study introduces a novel technique to forecast the candlestick price of Chinese stock market index by transforming the OHLC (Wang et al., 2021), as shown in equation 2.22 – 2.24. They applied this technique to vector auto-regression (VAR), and vector error correction (VEC). Moreover, they guarantee that the output of this technique does not violate the candlestick constraint. This study used MAPE, RMSE, and Accuracy ratio (average of the ratio of length intersection and union between actual and predicted value) to measure the performance of the model.



$$Y_t = \begin{bmatrix} y_t^{(1)} \\ y_t^{(2)} \\ y_t^{(3)} \\ y_t^{(4)} \end{bmatrix} = \begin{bmatrix} \ln(x_t^{(l)}) \\ \ln(x_t^{(h)} - x_t^{(l)}) \\ \ln\left(\frac{\lambda_t^{(o)}}{1 - \lambda_t^{(o)}}\right) \\ \ln\left(\frac{\lambda_t^{(c)}}{1 - \lambda_t^{(c)}}\right) \end{bmatrix} \quad (2.22)$$

$$\lambda_t^{(o)} = \frac{\exp\{y_t^{(3)}\}}{1 + \exp\{y_t^{(3)}\}} \quad (2.23)$$

$$\lambda_t^{(c)} = \frac{\exp\{y_t^{(4)}\}}{1 + \exp\{y_t^{(4)}\}} \quad (2.24)$$

The Convolutional Neural Network (CNN) model, which is well-suited to computer vision, is also used to predict the time series data. (Guo et al., 2018) predicted the next day price movement (direction) of Taiwan stock index by converting candlestick data (OHLC) to images and using CNN-Autoencoder to generate features for a 1D-CNN model to predict price movement. Another research recognized the candlestick pattern with Convolutional Neural Network (CNN) (Chen & Tsai, 2020). They also compared encoding OHLC price to image with Gramian Angular Field (GAF), which converts time series to polar coordinate, with encoding CULR price to the image. Moreover, they compared this technique with Long Short-Term Memory (LSTM). The result showed that using CULR with CNN to predict the candlestick pattern outperformed the other setting. The summary of this topic is shown in Table 6.

*Table 6 Studies relating to candlestick price and candlestick price prediction method*

Authors	Data	Output	Algorithm	Techniques	Measurement
(Alkhodhairi et al., 2021)	Bitcoin minute price but convert to 4h, 12h, 24h price (Jan 2017 to Aug 2020)	Next interval OHLC	Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU)	use OHLC price to predict OHLC price	- MAPE - RMSE - MAE - R2
(Wang et al., 2021)	Index of Chinese stock market	OHLC price	- Vector auto-regression (VAR), - Vector error correction (VEC)	use transformed OHLC price to predict transformed OHLC price	- MAPE - RMSE - Accuracy ratio
(Guo et al., 2018)	Taiwan stock index	next day price Movement (Up/Down)	Convolutional Neural Network (CNN)	- map candlestick data (OHLC) to image - CNN-Autoencoder to create feature for 1D-CNN model to predict price movement	- Accuracy - Precision - Recall - F1-Score

Authors	Data	Output	Algorithm	Techniques	Measurement
(Chen & Tsai, 2020)	EUR/USD 1-minute price (January 1, 2010, to January 1, 2018)	candlestick pattern	- Convolutional Neural Network (CNN) - LeNet - Long Short-Term Memory (LSTM)	Compare two methods - encoded <b>OHLC</b> price to image with Gramian Angular Field (GAF) - encoded <b>CULR</b> price to image with Gramian Angular Field (GAF)	Accuracy

From the above studies, there has been limited research on cryptocurrency candlestick price prediction. It was also found that RNNs generally outperformed MLP and traditional statistical models (such as ARIMA) in terms of forecasting error. Among the related work being investigated, the most popular normalization technique is the min-max normalization. There has been little research on cryptocurrency candlestick price prediction, and few studies have explored the performance of the candlestick price prediction method with recurrent neural networks in terms of both direction accuracy and forecasting error. Additionally, external factors do not appear to improve model forecasting performance.

Among the investigated studies, two studies were found to be the most similar to our work. The first study is (Alkhodhairi et al., 2021). They predicted the OHLC price of the next time period using the previous time period's OHLC price. They utilized LSTM and GRU models, which are similar to our work in terms of the model used. However, they did not investigate the effects of normalization on forecasting performance or the impact of different candlestick construction methods.

Another research similar to our work is (Jiang, 2020). The study focused on investigating the performance of two normalization approaches: whole set min-max normalization and sliding window relative change normalization on Bitcoin hourly price using LSTM, GRU and MLP model. However, they did not explore other normalization techniques such as z-score normalization or other combinations of normalization approaches (such as sliding window and whole set normalization) and techniques (such as min-max normalization). Additionally, they only investigated two combinations of normalization. Furthermore, the study did not investigate the reasons why sliding window normalization outperforms whole set normalization.

Therefore, this research aims to explore cryptocurrency candlestick price prediction using RNNs and simultaneously investigate various normalization techniques and their impact on RNNs performance. Additionally, this study aims to explore additional candlestick price prediction approaches that are not OHLC price prediction, such as candle wick (CULR) prediction

## Chapter 3: Methodology

### 3.1 Problem approach

Since the objective of this research is to predict the Bitcoin candlestick price of the next day and the Bitcoin historical trading data is time-series data, the suitable quantitative forecast method for this problem should be the time-series method or the machine learning method that the model can handle sequential data. Price prediction is a financial problem, so it requires high-accuracy prediction. In addition, the Bitcoin price is a high volatility trading data, so to precisely predict the price, complex models like neural networks, which are the machine learning method, are suitable to use to solve this problem instead of classical time series forecasting methods (McNally et al., 2018).

The neural network type which is suitable for time series forecasting is Recurrent Neural Networks (RNNs) because it is modified to handle sequence data. Based on the previous work in section 2, it is unclear whether LSTM or GRU perform better. As a result, both LSTM and GRU were used in all experiments in this work.

Our workflow is structured into three main phases: data preparation, feature construction, and model construction, as shown in Figure 4. The process begins with obtaining raw data, followed by cleansing the data, and performing feature selection and exploratory data analysis (EDA). In the feature construction phase, we split the data into three distinct sets, normalize the data, and create sliding windows. Then, we perform EDA once again. In the model construction phase, we train the models, tune their hyperparameters, and perform the inverse transformation or scaling back to the original scale. Finally, we evaluate the models' performance.

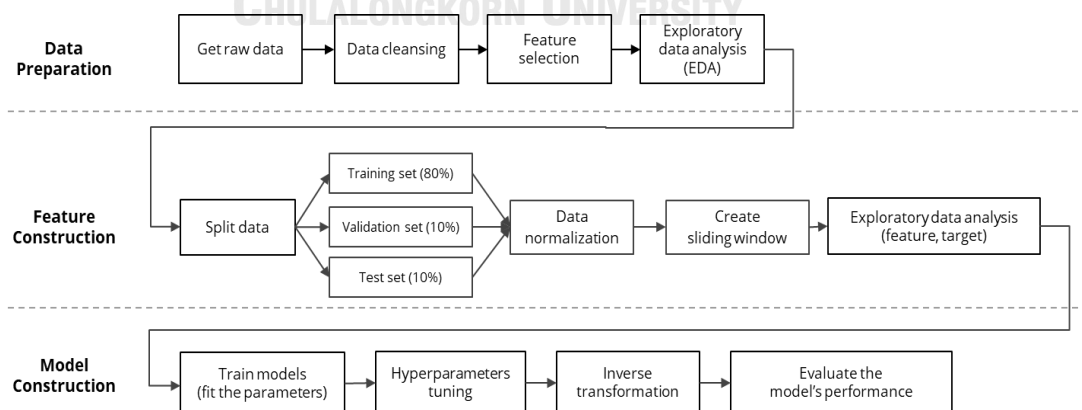


Figure 4 Workflow

### 3.1.1 Data source

This research uses historical trading data sets from Cryptodatadownload website (<https://www.cryptodatadownload.com>) of Bitcoin (BTC) and Tether coin (USDT), which is a stable coin where 1 USDT is approximately 1 USD. The

Cryptodatadownload website retrieves data from the Binance exchange, the world's largest cryptocurrency exchange by trading volume. The raw data from Cryptodatadownload includes ten columns, namely Unix timestamp, date, symbol,

opening price, highest price, lowest price, closing price, volume (Crypto), volume Base, and trade count. The meanings of each column are shown in Table 7. This study focuses on a one-day timeframe from August 17, 2017 to August 29, 2022 or 1,840 rows.

*Table 7 Raw data definitions*

Column	Definition
Unix timestamp	total of seconds that used to convert to local time zone
date	timestamp
symbol	symbol for which cryptocurrency converts to base coin
opening price	the time period's opening price
highest price	the time period's highest price
lowest price	the time period's lowest price
closing price	the time period's closing price
volume (Crypto)	volume in the transacted Crypto (BTC)
volume Base	volume in the base/converted crypto (USDT)
trade count	number of trades for the given time-period

\* BTC = Bitcoin, USDT = Tether coin which is a stable coin cryptocurrency and 1 USDT is approximate 1 USD

### 3.1.2 Data preparation

The steps of data preparation are as follows:

1. Data cleaning

Any empty rows and columns were eliminated. The date format was updated to be consistent across the entire dataset.

2. Feature selection

This work focuses on the candlestick prices (OHLC prices) which are open price, high price, low price, and close price.

3. Exploratory Data Analysis

This work used time series plots, decomposition plots, and histograms to analyze the data.

### 3.1.3 Feature construction

This study divided feature construction into three phases: Data splitting, normalization, and sliding window construction.

The steps of model construction are as follows:

#### 1. Data splitting

The data was divided into three sets: the training set contains 80% of the data for model training, the validation set contains 10% for hyperparameter tuning, and the test set contains 10% for model evaluation.

#### 2. Data normalization

From related work (chapter 2.1), the most popular normalization technique is min-max normalization, so we used this technique with the minimum and maximum values of the training set to normalize the data.

#### 3. Sliding window construction

The sliding window is a feature construction method that uses a rolling origin to separate data into small sets. This research used a fixed size sliding window, implying that each sliding window contains the same number of members which have some overlapped members. The sliding window of the OHLC price was used as the model's input in this experiment, as shown in equation 3.1, and the model's output is described in equation 3.2 which refers to the OHLC price of the next time step. Because of sliding window size effect on the forecasting performance (Tomar et al., 2022), different sliding window sizes give different errors, this study tested each treatment (algorithm and normalization technique) with various sliding window sizes which are 3, 5, 7, 15, and 30 days.

$$X_t = (x_t, x_{t+1}, x_{t+2}, \dots, x_{t+w-1}) \quad (3.1)$$

$$Y_t = (x_{t+w}) \quad (3.2)$$

Where  $x_t$  is vector of OHLC price at time step  $t$ ,  $w$  is sliding window size,  $X_t$  is input sliding window for model,  $Y_t$  is output for model.

### 3.1.4 Model construction

#### 3.1.4.1 Model configuration

The prediction of candlestick prices is a multivariate forecasting problem. Previous research (Du et al., 2019) demonstrated that employing a multiple-feature model outperforms using single-feature models, so this study used the multiple-feature model, as illustrated in Figure 5. Furthermore, the model architecture is Many-to-one, with  $x$  representing the feature's time

step,  $y$  representing the model's output, and  $a$  representing data transfer between time steps, as illustrated in Figure 6.

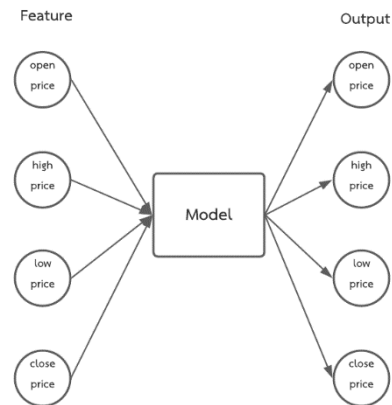


Figure 5 Feature diagram of the model

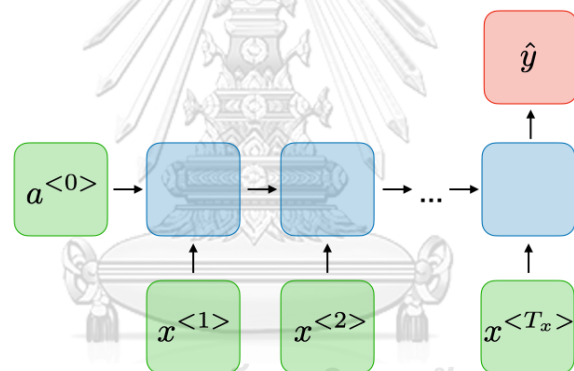


Figure 6 Structure diagram of the model (Afshine Amidi)

### 3.1.4.2 Model training and optimization

After the experiment settings have been determined, the models are trained and optimized by the following steps:

1. Train each model or algorithm on training set with MSE loss function
2. Hyperparameter tuning with Bayesian optimization by improving the validation set MSE and optimizing the hyperparameter as shown in Table 8. Note that we fixed the hyperparameter shown in Table 9.
3. Inverse transform the output from standardized data to the original representation (scale back)
4. Evaluate the performance with MAPE and RMSE on the test set

Table 8 The tuned hyperparameter and tuning range

Hyperparameter	Range	Step
Units	16 – 256	1
Dropout	0.2 – 0.5	0.1
Epoch	10 – 300	1
Batch size	32 - 256	Power of 2
Learning Rate	$10^{-4}$ - 1	log scale
epoch decay	10 - 50	10

Table 9 The fixed hyperparameter

Hyperparameter	Value	Note
Activation function	tanh	default of each architecture unit type
Output Activation function	Linear	default of each architecture unit type
Optimizer	Adam	Based on the previous work (Alkhodhairi et al., 2021)
Number of Layers	2	Based on the previous work (Alkhodhairi et al., 2021) (Jiang, 2020)
Decay rate	0.5	parameter for learning rate step decay

### 3.1.5 Tools and programming language

Python 3.7 is the main programming language used in this study, which runs on Google Colab. Furthermore, R 4.1 was used for exploratory data analysis (EDA). This project made use of several tools and Python libraries. For data pre-processing and modification, NumPy and Pandas were used. Matplotlib was used for graphical display of information and data, often known as data visualization. For data transformation, Scikit-learn was employed. TensorFlow and Keras were the primary libraries used for model training, prediction, and hyperparameter tuning.

### 3.2 Model performance improvement with normalization techniques

This part aimed to improve models using normalizing techniques. To achieve the objective of this study, this part compared two normalization approaches: whole set normalization and sliding window normalization. The whole set normalization means data are normalized by a training set parameter, and the sliding window normalization means the data are normalized by a sliding window parameter.

These two main normalization approaches led to a different process order in the feature construction phase. For the whole set approach, normalization occurs before sliding window construction, whereas normalization occurs after sliding window construction for the sliding window approach. The workflow of each normalization approach are shown in Figure 7.

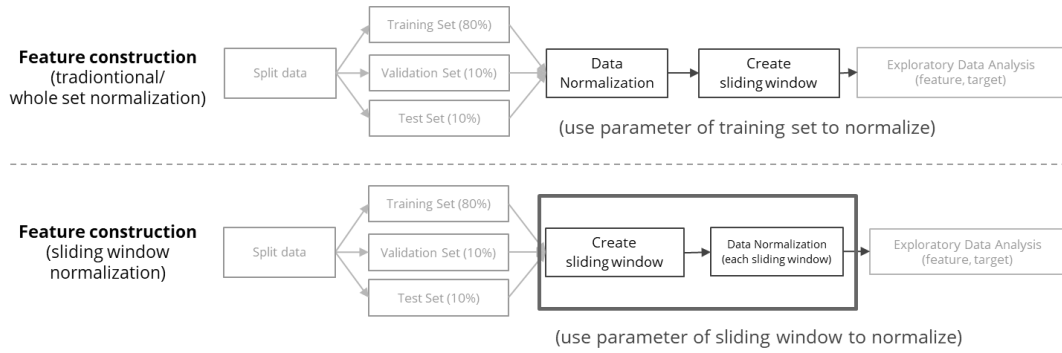


Figure 7 Workflow of each normalization approach

The investigated normalization techniques for both the whole set and the sliding window set include z-score normalization, min-max normalization, and relative change normalization. The details of all investigated normalizing techniques are shown in numbers 1–6 and equation 3.3-3.8.

Because of sliding window size effect on the forecasting performance (Tomar et al., 2022), different sliding window sizes give different errors, this study tested each treatment (algorithm and normalization technique) with various sliding window sizes which are 3, 5, 7, 15, and 30 days.

1. Z-Score normalization

To normalize the data, this technique uses the mean ( $\mu$ ) and standard deviation ( $\sigma$ ) of the original data as shown in equation 3.3. The training set's mean and standard deviation were utilized for whole-set z-score normalization.

$$x' = \frac{x - \mu}{\sigma} \quad (3.3)$$

2. Min-Max normalization

As demonstrated in equation 3.4, the minimum value ( $x_{\min}$ ) and maximum value ( $x_{\max}$ ) of the original data are used to linear transform the data. The training set's minimum and maximum are utilized to transform the training set and the other sets for the whole set min-max normalization.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3.4)$$

3. Relative change normalization

The relative change normalizes the data using the first value of the training set ( $x_0$ ) as illustrated in equation 3.5.

$$x' = \frac{x - x_0}{x_0} \quad (3.5)$$



#### 4. Window z-score normalization

Unlike the whole set normalization, the window z-score normalization used the mean ( $\mu_w$ ) and standard deviation ( $\sigma_w$ ) of the sliding window with as shown in 3.6.

$$x'_i = \frac{x_i - \mu_w}{\sigma_w} \quad (3.6)$$

#### 5. Window min-max normalization

This technique also uses the statistic of the sliding window which are the minimum value of the window ( $x_{\min W}$ ) and maximum value of the window ( $x_{\max W}$ ) as illustrated in equation 3.7.

$$x'_i = \frac{x_i - x_{\min W}}{x_{\max W} - x_{\min W}} \quad (3.7)$$

#### 6. Window relative change normalization

The relative change normalizes the window elements by the first value of the sliding window ( $x_0$ ) as illustrated in equation 3.8.

$$x'_i = \frac{x_i - x_0}{x_0} \quad (3.8)$$

### 3.3 Candlestick price prediction method

This section aims to compare the performance of candlestick price prediction methods in terms of both direction accuracy and forecasting error. The investigated methods include using OHLC price to predict OHLC price and using CULR price to predict CULR price.

OHLC price or candlestick price means four prices in the specific time interval which contains the opening price, the highest price, the lowest price, and the closing price of that time interval, as shown in Figure 8.

Another way to display the candlestick price is CULR. CULR or candlewick price also contains four components: closing price, upper shadow, lower shadow, and real body. The visualization of CULR is also shown in Figure 8. The CULR method includes additional processes compared to the OHLC method, specifically, the conversion of OHLC features to CULR features, as shown in Figure 9. This conversion can be accomplished using equations 3.9 - 3.11, and CULR features can be converted back to OHLC features by equations 3.12 - 3.14.

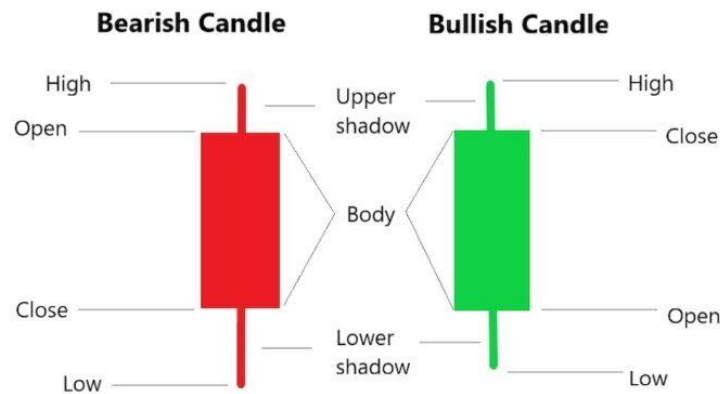


Figure 8 Candlestick component (16 must-know candlestick patterns for a successful trade, 2021)

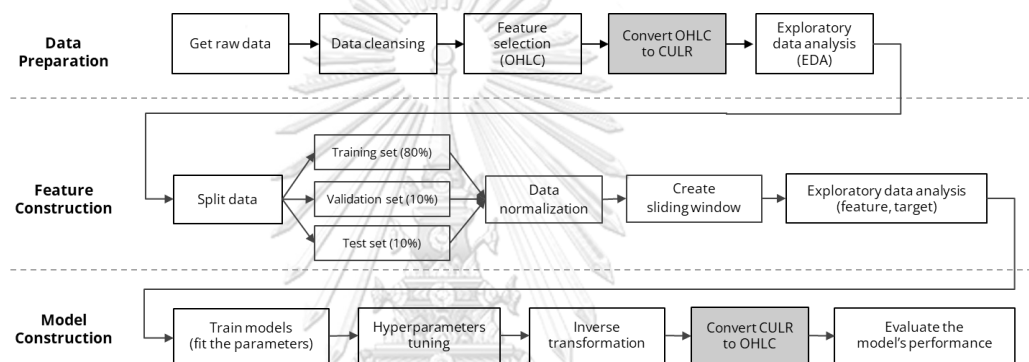


Figure 9 Workflow of CULR method

$$\text{Real Body} = \text{close} - \text{open} \quad (3.9)$$

$$\text{Upper Shadow} = \begin{cases} \text{high} - \text{close} & \text{Real Body} \geq 0 \text{ (Bullish)} \\ \text{high} - \text{open} & \text{Real Body} < 0 \text{ (Bearish)} \end{cases} \quad (3.10)$$

$$\text{Lower Shadow} = \begin{cases} \text{open} - \text{low} & \text{Real Body} \geq 0 \text{ (Bullish)} \\ \text{close} - \text{low} & \text{Real Body} < 0 \text{ (Bearish)} \end{cases} \quad (3.11)$$

$$\text{open} = \text{close} - \text{Real Body} \quad (3.12)$$

$$\text{high} = \begin{cases} \text{close} + \text{Upper Shadow} & \text{Real Body} \geq 0 \text{ (Bullish)} \\ \text{open} + \text{Upper Shadow} & \text{Real Body} < 0 \text{ (Bearish)} \end{cases} \quad (3.13)$$

$$\text{low} = \begin{cases} \text{open} - \text{Lower Shadow} & \text{Real Body} \geq 0 \text{ (Bullish)} \\ \text{close} - \text{Lower Shadow} & \text{Real Body} < 0 \text{ (Bearish)} \end{cases} \quad (3.14)$$

We compared the methods by utilizing several normalization techniques that were investigated in Section 3.2. Moreover, we evaluated each method using various sliding window sizes, including 3, 5, 7, 15, and 30 days. The performance of candlestick price prediction methods

using the OHLC price was compared. For the CULR method, we transformed the predicted CULR price into OHLC price format.

To compare direction accuracy, we converted the predicted OHLC price (numeric) to direction (categorical) or price movement: bullish (upward) and bearish (downward). Note that for neutral direction or when the close price is equal to the open price, it is treated as bullish. The equation for direction conversion is shown in equation 3.15. The performance metric for forecasting error is RMSE, and the performance metric for direction is accuracy.

$$Direction = \begin{cases} \text{Bullish} & close \geq open \\ \text{Bearish} & close < open \end{cases} \quad (3.15)$$



## Chapter 4: Result and discussion

### 4.1 Exploratory data analysis

#### 4.1.1 Exploratory data analysis before data splitting

After selecting the features, the data was explored through a time series plot depicted in Figure 10, summary statistics presented in Table 10, and decomposition analysis illustrated in Figure 11.

According to Figure 10, each candlestick price or OHLC price had the same major feature but was different in detail that was not significant for overview consideration, so in this section, the open price is representative of the OHLC price to analyze the data character. Although the price fluctuated from 2017 to the end of 2020, it did not vary significantly when compared to the cryptocurrency boom period from the end of 2020 to 2021, which was a clear uptrend. The price had been dropping from the end of 2021 to the beginning of 2022.

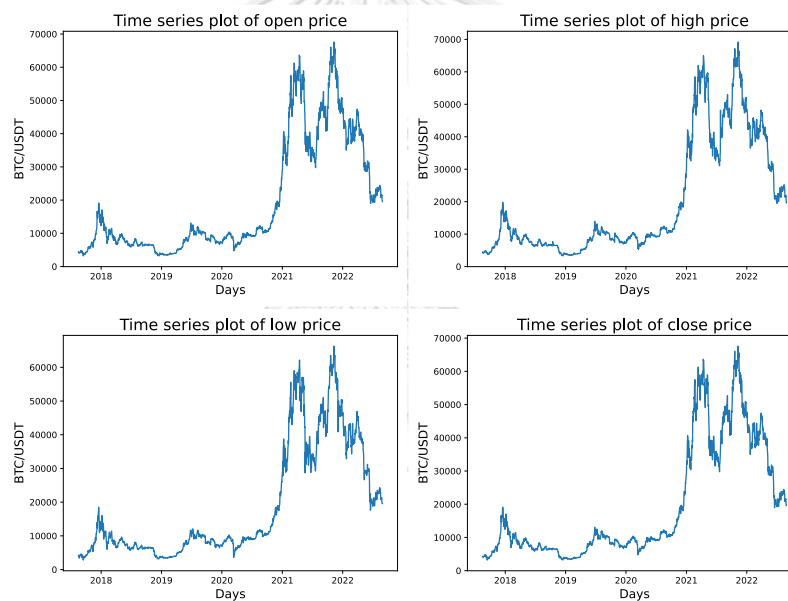


Figure 10 Time series plot of OHLC prices.

The following step is to find a basic statistical summary of each price. Table 10 displays the following fundamental statistics: minimum, first quartile, median, mean, third quartile, and maximum.

This study also used decomposition to determine the data's component. The open price was chosen to represent the OHLC prices since they have similar properties. The price in this dataset has three components, as illustrated in Figure 11. They are trend or trend-cyclical, seasonal, and irregular, and they are joined by multiplicative decomposition.

Table 10 Basic statistical summary table of each price

Statistics	Open price	High price	Low price	Close price
count	1,840	1,840	1,840	1,840
mean	19,484.23	20,057.92	18,836.06	19,494.10
std	17,284.92	17,778.82	16,722.02	17,281.56
min	3,189.02	3,276.50	<b>2,817.00</b>	3,189.02
max	67,606.96	<b>69,198.70</b>	66,300.00	67,606.96

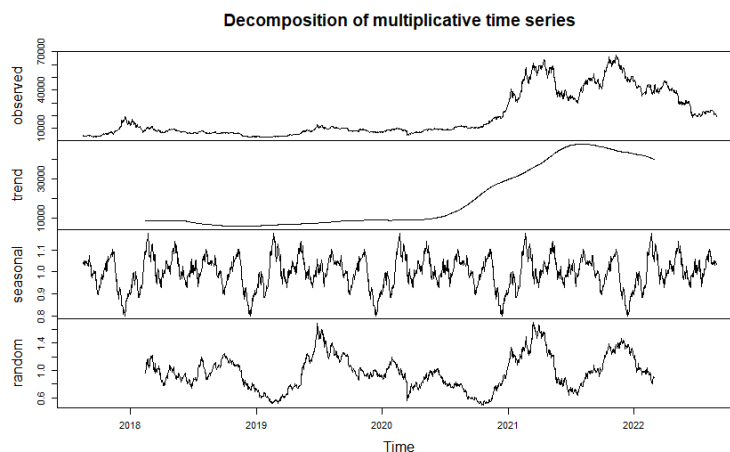


Figure 11 The decomposition plot of open price

#### 4.1.2 Exploratory data analysis after data splitting

After splitting the data into three sets, the data was explored by the time series plot and histogram, as shown in Figure 12-13 respectively. According to part 4.1.1, the open price is representative of the OHLC price to analyze the data character. In addition, this part also shows a basic statistical summary of the open price after splitting the data in Table 11.

Table 11 Basic statistical summary table of open price after data splitting

Set	Size (rows)	Percentage (%)	Mean	Std	Min	Max	Skewness	Skewness interpretation
Training set	1472	80	14,315.33	14,123.20	3,189.02	63,645.05	1.147	Highly skewed
Validation set	184	10	49,474.80	8,314.67	35,043.73	67,606.96	0.652	Moderately skewed
Test set	184	10	30,844.89	8,992.62	18,970.79	47,418.5	0.418	Approximately symmetric
Whole data set	1840	100	19,484.23	17,284.92	3,189.02	67,606.96	1.623	Highly skewed

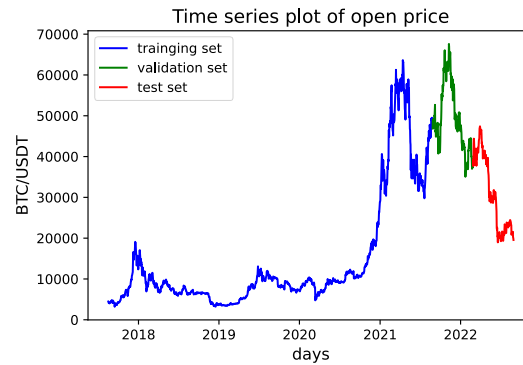


Figure 12 Time series plot of open prices with data splitting

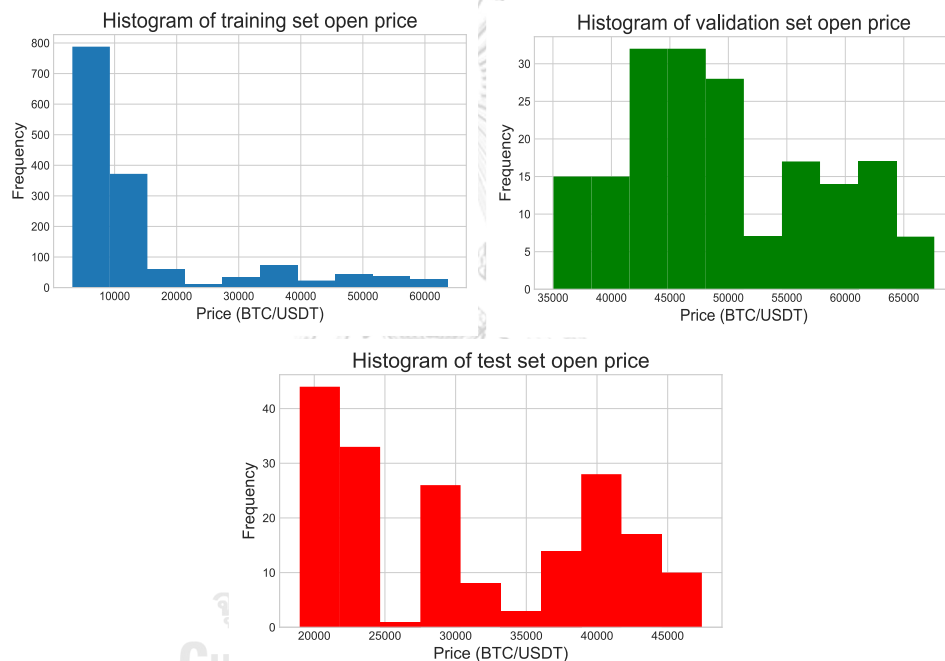


Figure 13 Distribution of open price with data splitting

## 4.2 Model performance

This section shows the forecasting performance of the LSTM and GRU, which were averaged across various sliding window sizes of 3, 5, 7, 15, and 30 days. Table 12 shows the model performance in terms of MAPE, which ranges between 4.29% - 7.92%. Overall, LSTM has an average MAPE over the OHLC prices of 6.99%, while GRU's MAPE is 4.96%. Table 13 shows the performance in terms of RMSE, ranging from 1,603.14 – 2,729.23. Overall, RMSE of LSTM is 2,458.21, and RMSE of GRU is 1,857.86. Therefore, the forecasting errors of the GRU are generally lower than the errors of the LSTM in our case.

Table 12 Average and standard deviation of MAPE (%) on test set for LSTM and GRU.

Algorithm	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
LSTM	6.03	2.07	7.29	2.53	6.72	LSTM	6.03	2.07	7.29	2.53
GRU	4.29	1.01	5.30	0.93	4.73	GRU	4.29	1.01	5.30	0.93

Table 13 Average and standard deviation of RMSE on test set for LSTM and GRU.

Algorithm	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
LSTM	2162.40	636.56	2514.33	736.78	2426.87	630.73	2729.23	783.55	2458.21	675.31
GRU	1603.14	294.81	1925.27	269.93	1838.87	276.33	2064.17	251.53	1857.86	304.30

### 4.3 Model performance improvement with various normalization techniques

To evaluate the effect of normalization methods on forecasting performance, LSTM and GRU models were trained and tested on the normalized data with various sliding window sizes. The average and standard deviation of MAPE and RMSE for the OHLC prices over various sliding window sizes of LSTM are shown in Table 14-15 and GRU in Table 16-17, respectively. Furthermore, the overall average RMSE of each normalization technique and model algorithm is also illustrated in Figure 14, and each price average RMSE is presented in Figure 15.

From Table 14-17, note that all sliding window normalization techniques (the last three rows) yield significantly lower errors than the whole set normalization techniques (the first three rows) for both LSTM and GRU. Moreover, the window relative change normalization yields the lowest average errors in terms of MAPE and RMSE for both LSTM and GRU techniques in most cases. Nonetheless, since the errors of all three sliding window normalization techniques are relatively close, it is not obvious if any sliding window normalization technique definitely dominates the others. However, it can be concluded that the sliding window normalization techniques, in general, significantly outperform the whole set normalization techniques in terms of forecasting performance.

Table 14 LSTM average and standard deviation of MAPE (%) on the test set

Normalization Technique	open		high		low		close		overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set z-score	6.13	1.69	7.51	2.52	5.46	1.44	7.54	2.00	6.66	2.02
Whole set min-max	6.03	2.07	7.29	2.53	6.72	2.20	7.92	2.80	6.99	2.33
Whole set relative change	11.43	3.38	12.36	3.89	10.97	2.79	11.98	3.42	11.69	3.16
Window z-score	1.98	1.17	2.13	0.20	2.50	0.36	2.96	0.20	2.40	0.69
Window min-max	2.04	1.16	2.10	0.13	2.31	0.22	2.67	0.10	<b>2.28</b>	0.60
Window relative change	1.47	0.50	2.31	0.33	2.44	0.33	3.00	0.31	2.31	0.66

Table 15 LSTM average and standard deviation of RMSE on the test set

Normalization technique	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set z-score	2054.90	347.37	2485.50	599.22	1958.31	298.74	2552.45	417.41	2262.79	476.83
Whole set min-max	2162.40	636.56	2514.33	736.78	2426.87	630.73	2729.23	783.55	2458.21	675.31
Whole set relative change	3702.02	957.95	4079.99	1135.40	3556.16	784.95	3975.25	975.30	3828.35	916.84
Window z- score	953.24	711.31	825.90	104.79	989.94	73.21	1172.31	71.33	985.35	356.64
Window min-max	989.43	720.34	818.38	78.31	948.23	56.78	1094.38	43.59	962.60	349.14
Window relative change	578.06	168.41	862.01	89.78	965.24	85.99	1190.87	76.72	<b>899.04</b>	247.99

Table 16 GRU average and standard deviation of MAPE (%) on the test set

Normalization Technique	open		high		low		close		overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set Z-Score	5.62	1.84	6.42	1.30	4.91	0.41	6.40	0.52	5.84	1.25
Whole set Min-Max	4.29	1.01	5.30	0.93	4.73	0.97	5.53	0.94	4.96	1.02
Whole set relative change	11.36	6.30	11.89	6.42	10.64	5.18	11.93	5.24	11.46	5.36
Window Z-Score	1.84	1.28	2.14	0.34	2.42	0.21	2.96	0.47	2.34	0.78
Window Min-Max	1.93	1.16	2.06	0.14	2.35	0.18	2.74	0.19	2.27	0.64
Window Relative change	1.46	0.45	2.25	0.28	2.37	0.29	2.94	0.24	<b>2.25</b>	0.62



Table 17 GRU average and standard deviation of RMSE the test set

Normalization technique	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set z-score	2277.30	652.74	2627.58	488.66	2091.83	190.67	2625.52	250.85	2405.56	465.62
Whole set min-max	1603.14	294.81	1925.27	269.93	1838.87	276.33	2064.17	251.53	1857.86	304.30
Whole set relative change	3906.07	2073.65	4225.82	2119.07	3682.77	1590.14	4075.87	1684.74	3972.63	1738.81
Window z-score	912.87	784.69	845.45	172.52	992.33	78.44	1212.84	232.24	990.87	410.71
Window min-max	980.86	782.37	815.17	89.31	942.34	40.62	1098.85	61.98	959.31	377.48
Window relative change	560.21	153.14	834.16	62.85	930.75	72.73	1156.94	45.58	<b>870.52</b>	235.69

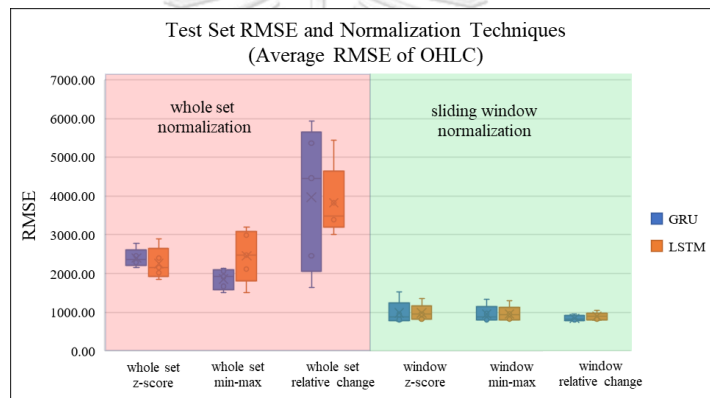


Figure 14 Box plot showing average RMSE of OHLC with various normalization techniques in GRU and LSTM

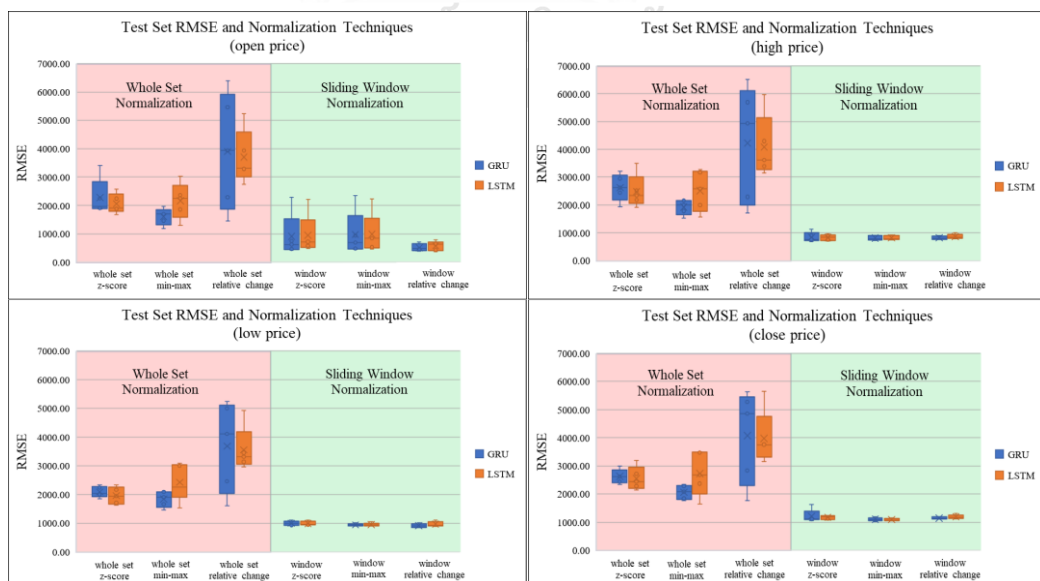


Figure 15 Box plot showing each price RMSE with various normalization techniques in GRU and LSTM

#### 4.4 Impact of normalization on data characteristics

This section explores why sliding window normalization techniques seem to outperform whole set normalization techniques by observing data characteristics, particularly the model input distribution and the sliding window for model input.

##### 4.4.1 Impact of normalization on data distribution

This section explores why the sliding window normalization techniques seem to outperform the whole set normalization ones by observing the skewness of data distributions, especially the model input distribution.

Figure 16a-16c show a sample of open-price input distributions of the training, validation, and test sets with a sliding window size of 7 days before normalization. The input distributions investigated in this section are derived from the sliding window of the model input, which is combined into a single column. Note that the input distribution of each set differs. The training set is highly skewed. The validation and test sets are both moderately skewed.

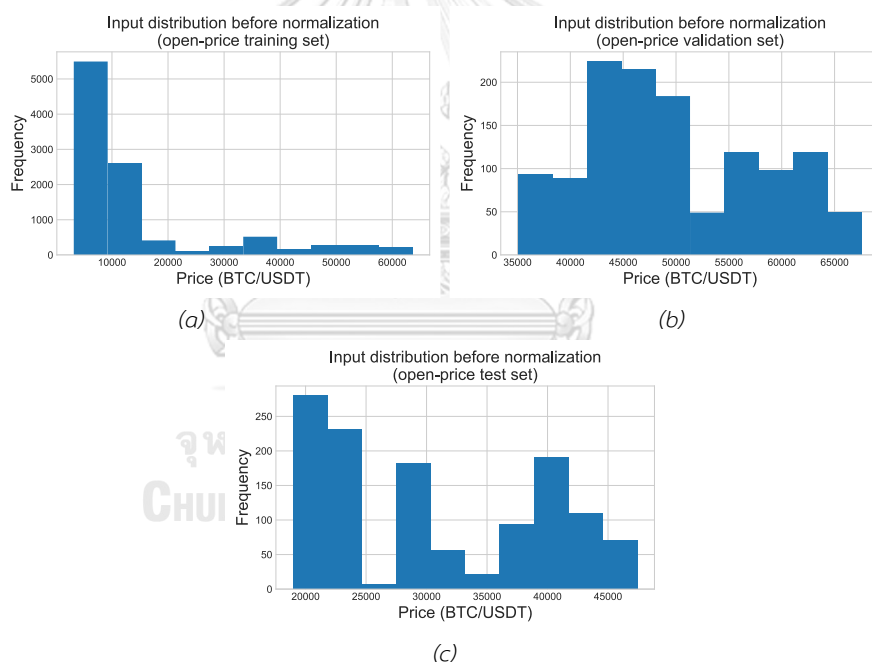


Figure 16 Sample input distribution before normalization of (a) training set (b) validation set (c) test set

Tables 18-20 illustrate the skewness of data distribution after various normalization techniques on the training set, the validation set and the test set, respectively. The skewness values shown in the table are root mean square (RMS) values of five skewness values when the window sizes are 3, 5, 7, 15, 30 days to provide an overall view of skewness across various window sizes. Additionally, sample box plots of open price skewness (magnitude) for the training set, validation set, and test set of input distribution with various normalization techniques are shown in Figures 17-19.

Table 18 Average of training set input distribution skewness

Normalization Technique	Open	High	Low	Close
Without normalization	1.136	1.134	1.120	1.139
Whole set z-score	1.136	1.134	1.120	1.139
Whole set min-max	1.136	1.134	1.120	1.139
Whole set relative change	1.136	1.134	1.120	1.139
Window z-score	0.053	0.099	0.057	0.054
Window min-max	0.047	0.110	0.055	0.051
Window relative change	0.333	0.375	0.291	0.334

Table 19 Average of validation set input distribution skewness

Normalization Technique	Open	High	Low	Close
Without normalization	0.676	0.645	0.594	0.678
Whole set z-score	0.676	0.645	0.594	0.678
Whole set min-max	0.676	0.645	0.594	0.678
Whole set relative change	0.676	0.645	0.594	0.678
Window z-score	0.084	0.136	0.050	0.084
Window min-max	0.120	0.164	0.110	0.116
Window relative change	0.141	0.151	0.117	0.145

Table 20 Average of test set input distribution skewness

Normalization Technique	Open	High	Low	Close
Without normalization	0.421	0.378	0.422	0.425
Whole set z-score	0.421	0.378	0.422	0.425
Whole set min-max	0.421	0.378	0.422	0.425
Whole set relative change	0.421	0.378	0.422	0.425
Window z-score	0.132	0.152	0.040	0.130
Window min-max	0.137	0.144	0.081	0.144
Window relative change	0.455	0.456	0.457	0.472

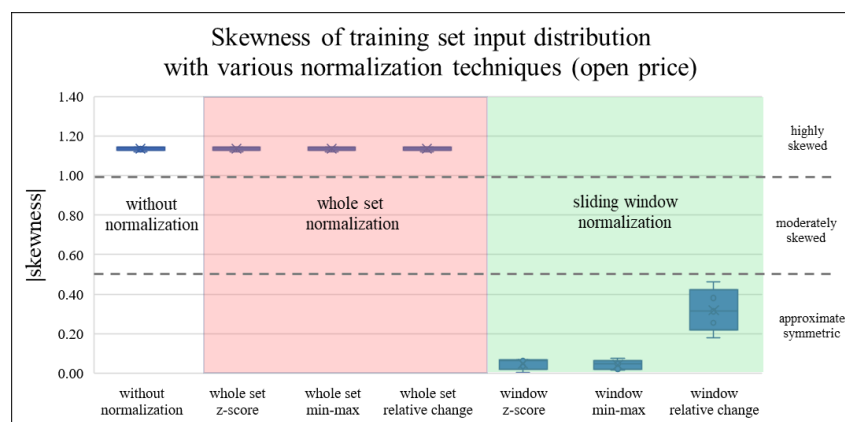


Figure 17 Skewness of training set open price input distribution with various normalization techniques

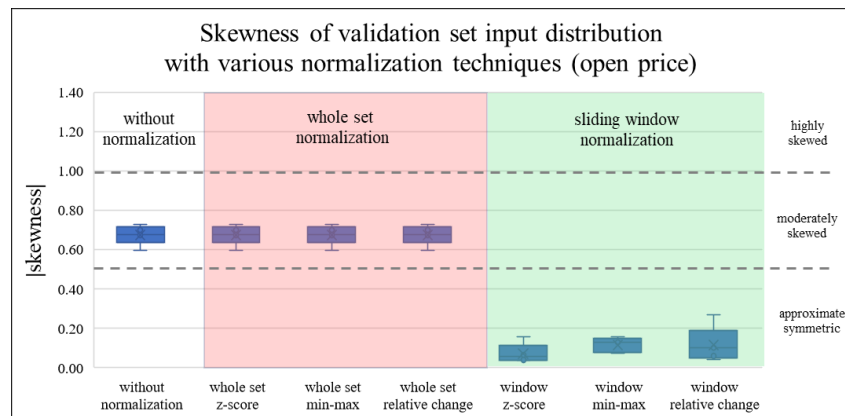


Figure 18 Skewness of validation set open price input distribution with various normalization techniques

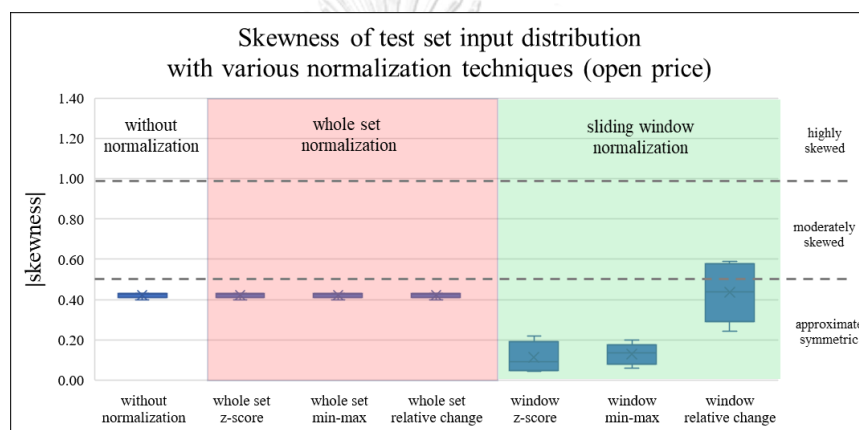


Figure 19 Skewness of test set open price input distribution with various normalization techniques

Table 18 displays the training set skewness before and after each normalization technique. Note that the data before normalization are highly skewed. Moreover, the whole set normalization techniques do not affect the distribution in terms of skewness at all. On the contrary, the skewness of the data distribution are all lower after the sliding window normalization techniques. In fact, after all three window normalization techniques, the data distributions are close to symmetric because their skewness values are between -0.5 and 0.5. Samples of transformed data distributions after various normalization techniques are shown in Figure 20a-20f. Note that all three sliding window normalization techniques help decrease skewness of highly skewed data.

Table 19 and 20 show similar impact on data distributions after normalization on the validation set and the test set. Again, the whole set normalization techniques do not change the distribution skewness, while the sliding window techniques help reduce the skewness for all distributions to almost symmetric. However, we note that the window relative change

normalization technique seems to have less impact in reducing the skewness, especially on the test set, when compared to the window z-score and the window min-max techniques.

The results in this section illustrate that the sliding window normalization techniques generally lead to more symmetric (less skewness) in the input data distribution than the whole set normalization. Potentially due to this impact, more symmetric input distributions then lead to better forecasting performances in both LSTM and GRU models.

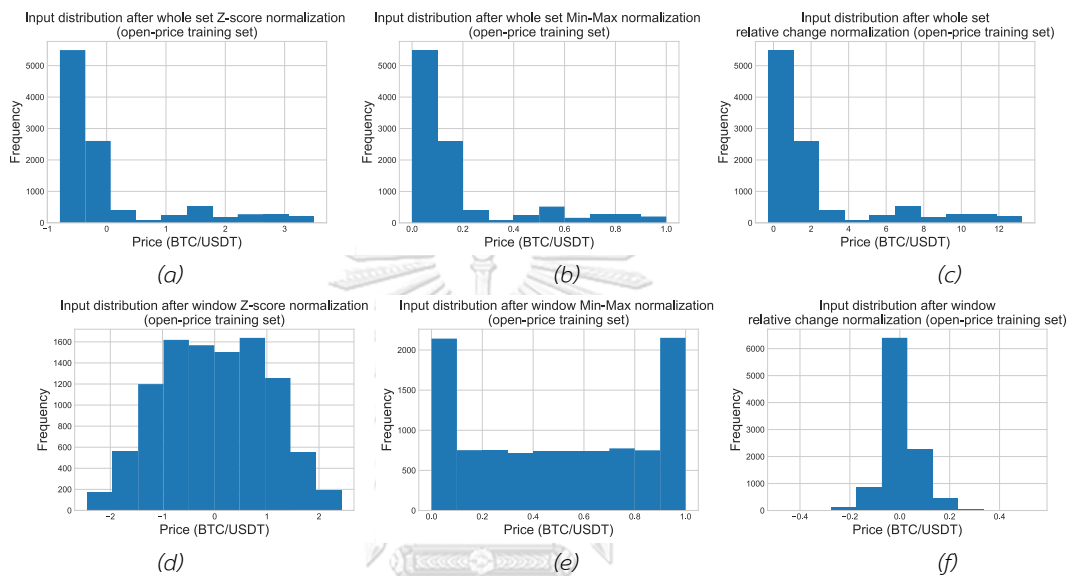


Figure 20 Sample input distribution after normalization by (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change

#### 4.4.2 Impact of normalization on sliding window

In order to provide a clearer understanding of why sliding window normalization techniques seem to outperform whole set normalization, we also examined the statistical properties of sliding windows by plotting the mean and standard deviation of the sliding windows over time. Figure 21 displays samples of the sliding window mean for the open price using various normalization techniques with a window size of 7 days. Figure 22 shows samples of the sliding window standard deviation for the open price using the same techniques and window size. Note that sliding window mean and sliding window standard deviation for whole set normalization are equivalent to moving average and moving standard deviation, respectively.

According to Figures 21-22, it was found that sliding window statistical properties of normalized data using sliding window normalization in each sliding window have a stable characteristic, as observed by the value of sliding window mean and sliding window standard deviation that do not significantly change over time. Additionally, the explored statistical properties in each data set have similar values. However, in the case of whole set normalization,

sliding window properties vary considerably over time, and the sliding window properties of each data set are significantly different, particularly for sliding window mean.

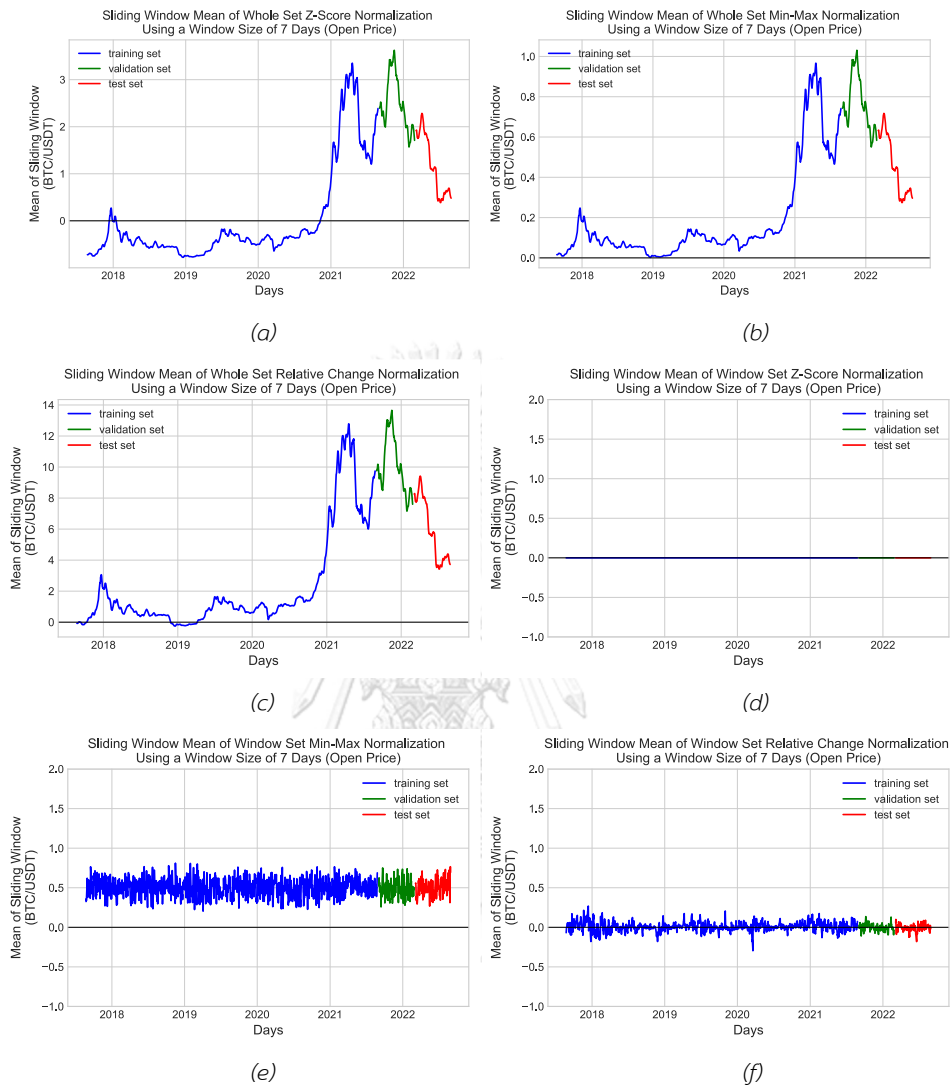
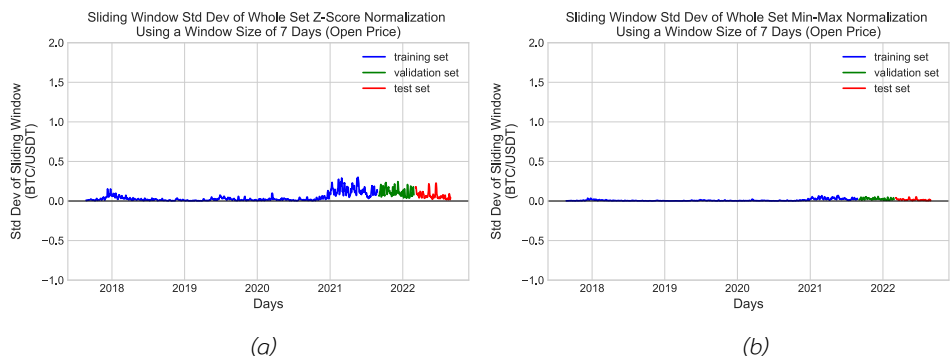


Figure 21 Sliding window mean after normalization by (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change



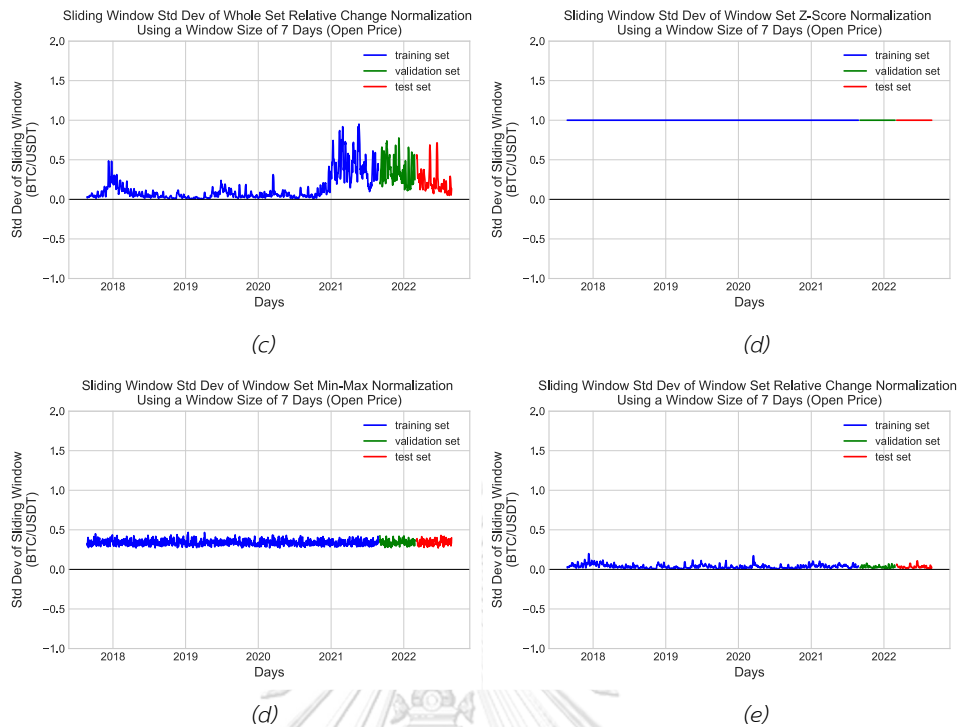


Figure 22 Sliding window standard deviation after normalization by (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change

To further elaborate the findings presented in Figures 21 and 22, we investigated the characteristics of three sliding windows from each dataset using all the normalization techniques under consideration. In addition, we used open price as representative of OHLC. Figure 23 presents a time series plot of the open price, along with selected sliding windows for observation. We selected an example of nine sliding windows including different characteristics (moving upward, moving downward, and moving sideways), with the specified time range for each window provided in Table 21.

Figures 24-27 showcase time series plots of the open price sliding window for the model input using various normalization techniques at each observation window with a sliding window size of 7 days. These include without normalization, z-score normalization, min-max normalization, and relative change normalization. Additionally, Figures 25-27 display both the whole set normalization approach and the sliding window normalization approach.

Figures 28a-28d present box plots of open price sliding window with various normalization at each observation window. These also include without normalization, z-score normalization, min-max normalization, and relative change normalization. For Figures 28b-28d, each normalization technique is shown with both the whole set normalization approach and the sliding window normalization approach.

According to Figures 25 – 27 and Figures 28b – 28d, we observed that the sliding window of the whole set normalization technique showed a significant difference in the range of each observation sliding window, particularly across the dataset. Conversely, the range of each observation point under the sliding window normalization technique was found to be relatively similar.

The findings of this section indicate that the sliding window normalization techniques have more stable sliding window statistical characteristics. This is evidenced by the sliding window mean, standard deviation, and range of each sliding window used as input for the model. These stable characteristics lead to a similar sequence for prediction, which ultimately results in better forecasting performance in both LSTM and GRU models when using sliding window normalization.

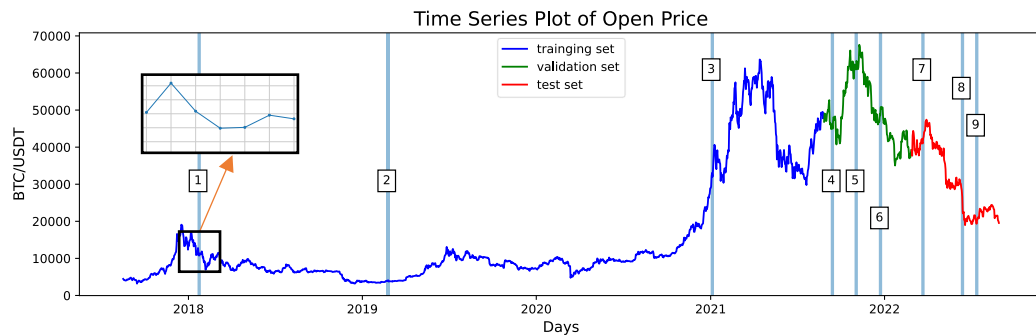


Figure 23 Time series plot of open price with selected sliding windows for observation

Table 21 Time range of the observation windows

Number	Set	Start date	End date
1	Training set	2018-01-20	2018-01-26
2		2019-02-20	2019-02-26
3		2021-01-01	2021-01-07
4	Validation set	2021-09-10	2021-09-16
5		2021-10-30	2021-11-05
6		2021-12-20	2021-12-26
7	Test set	2022-03-19	2022-03-25
8		2022-06-10	2022-06-16
9		2022-07-10	2022-07-16



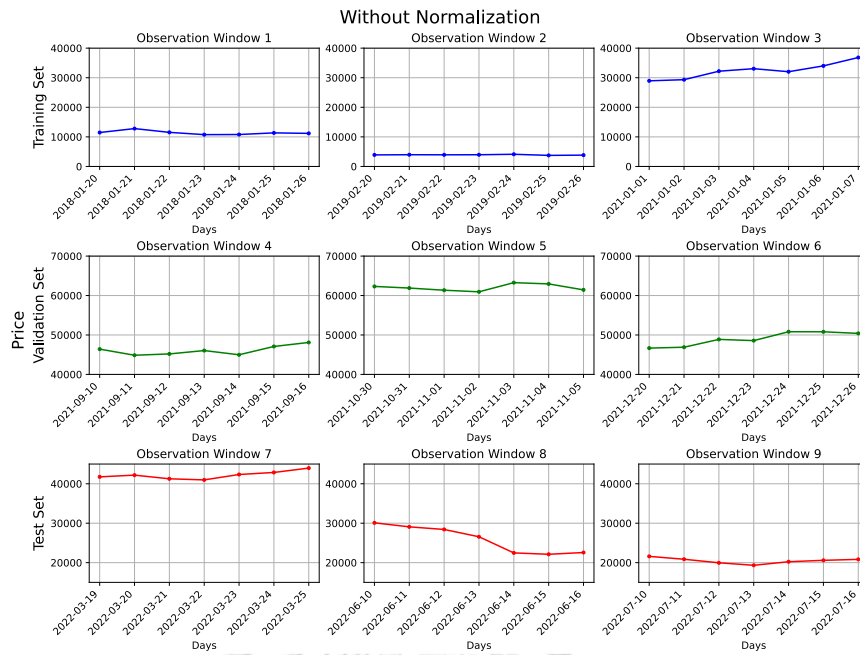


Figure 24 Time series plot of open price sliding window without normalization at each observation window

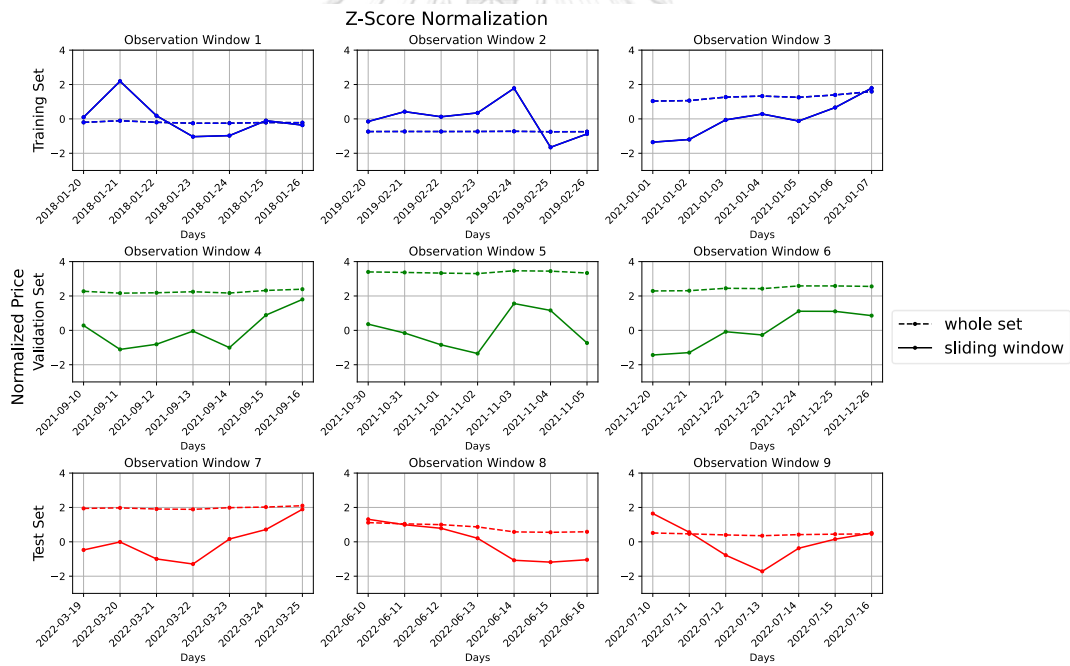


Figure 25 Time series plot of open price sliding window with z-score normalization techniques at each observation window

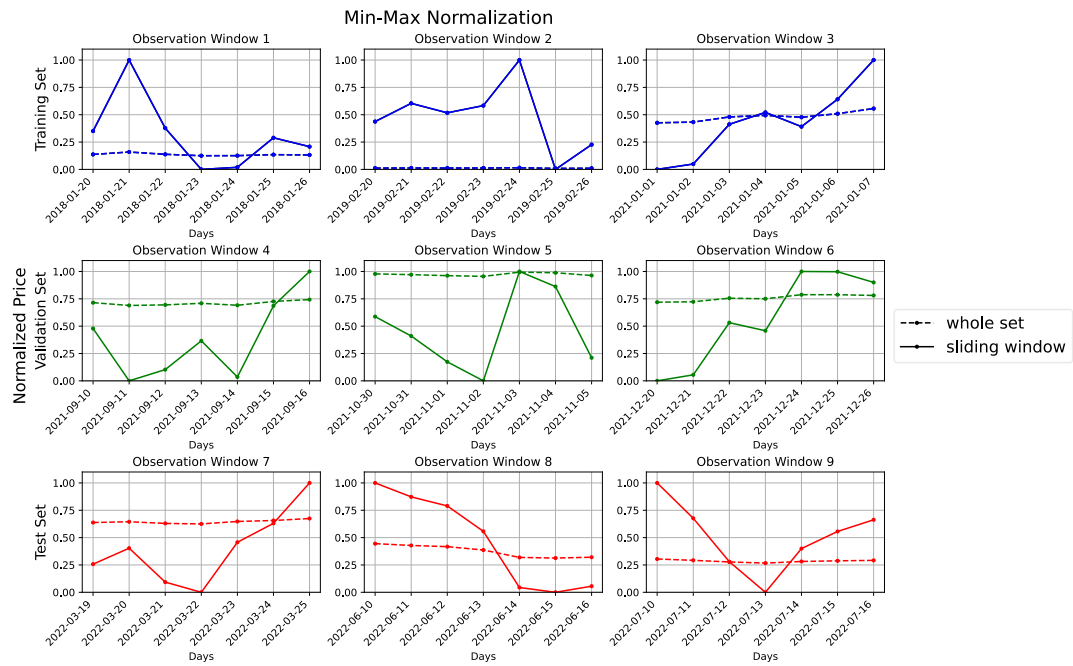


Figure 26 Time series plot of open price sliding window with min-max normalization techniques at each observation window

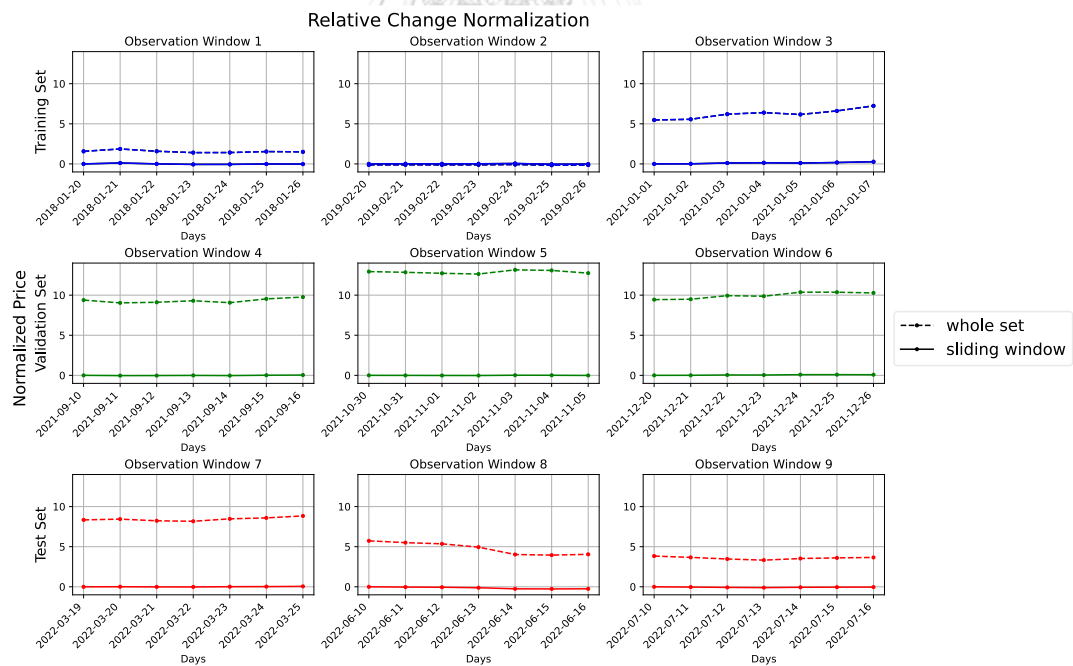


Figure 27 Time series plot of open price sliding window with relative change normalization techniques at each observation window

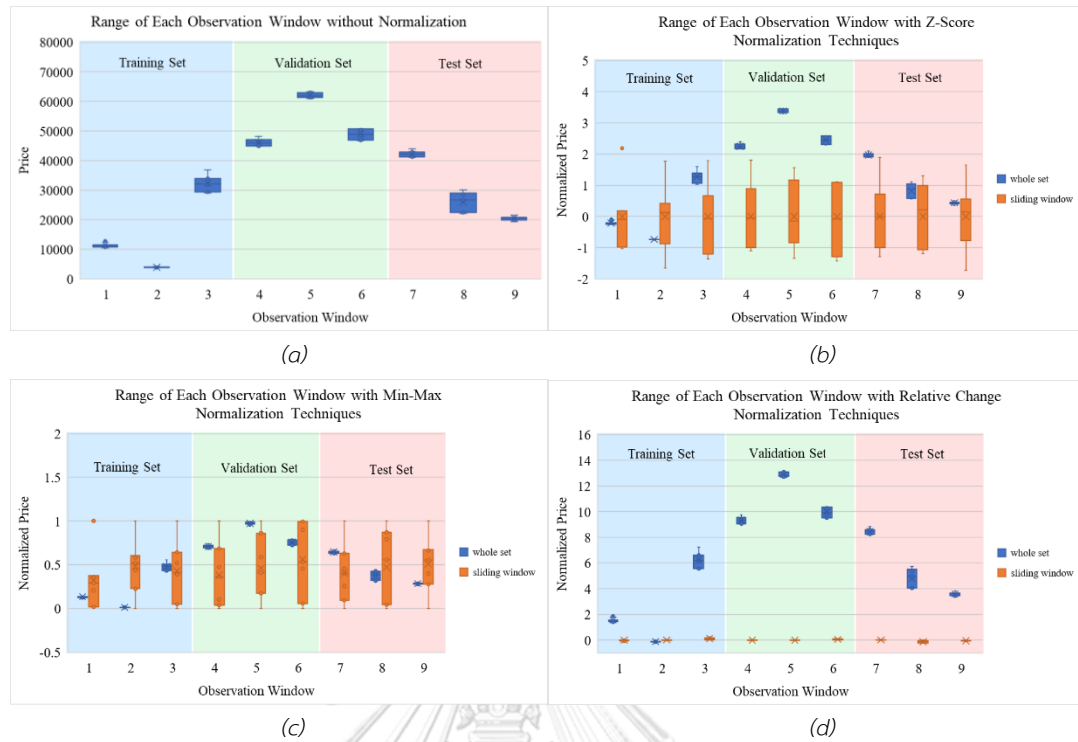


Figure 28 Box plot of open price sliding window with various normalization at each observation window a) without normalization b) z-score normalization c) min-max normalization d) relative change normalization

In summary, our analysis reveals the following findings. Firstly, the sliding window normalization techniques tend to result in a more symmetric input data distribution, as compared to the whole set normalization technique. This increased symmetry potentially explains the improved forecasting performance observed in both the LSTM and GRU models. Secondly, the sliding window normalization approach ensures that the models observe a similar sequence for prediction, which contributes to superior predictive performance when compared to models using the whole set normalization approach.

## 4.5 Exploring candlestick price prediction method

This section compares model performance with different candlestick construction methods: OHLC method and CULR method. Note that the results of OHLC are shown in section 4.1 - 4.3.

### 4.5.1 Exploring the model with CULR feature

#### 4.5.1.1 Exploratory data analysis for CULR

After converting OHLC features to CULR features, the data was explored through a time series plot, as illustrated in Figure 29a-29d, and a basic statistical summary for each feature was generated and presented in Table 22. Our analysis revealed that the characteristics of each feature were not similar, which differs from the OHLC feature. In addition, we present the decomposition plot of close price, upper shadow, lower shadow, and real body in Figures 30 to 33, respectively.

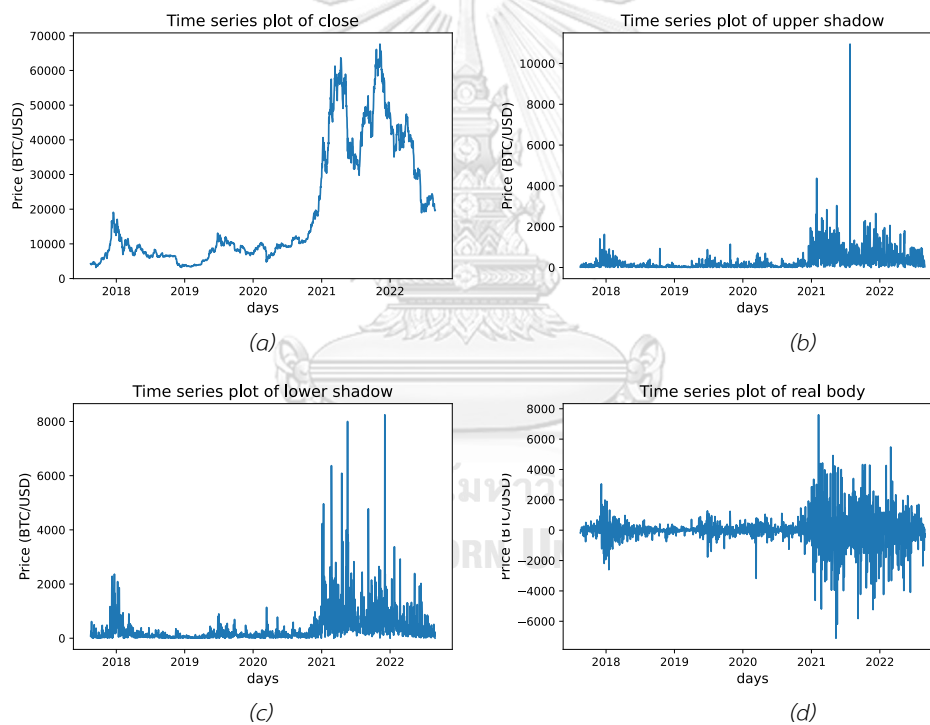


Figure 29 Time series plot of CULR (a) close price (b) upper shadow (c) lower shadow (d) real body

Table 22 Summary statistics of each feature

Statistical property	Close	Upper shadow	Lower shadow	Real body
count	1840	1840	1840	1840
mean	19,494.10	289.56	373.9	9.87
std	17,281.56	478.46	628.56	1,016.82
min	3,189.02	0	0	-7,116.94
max	67,606.96	10,950.51	8,246.77	7,602.08

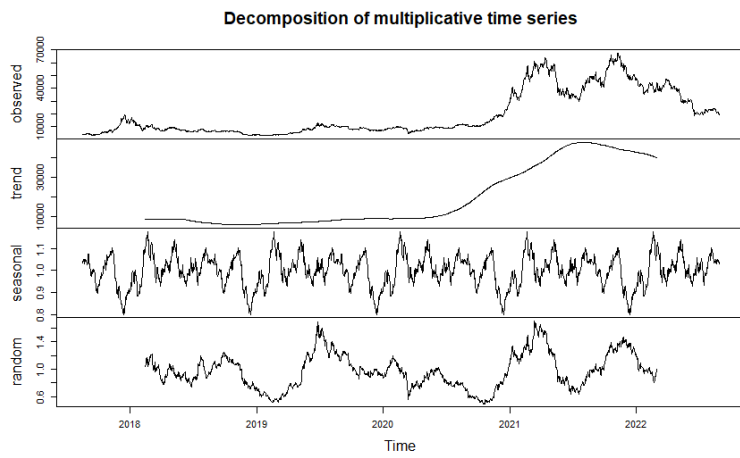


Figure 30 The decomposition plot of close price

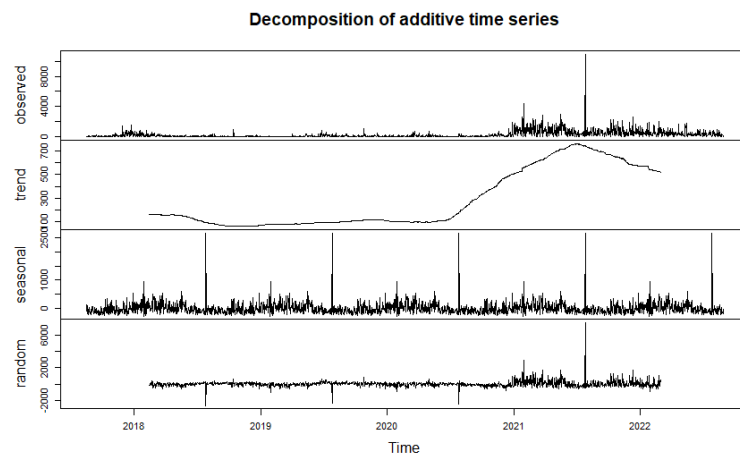


Figure 31 The decomposition plot of upper shadow

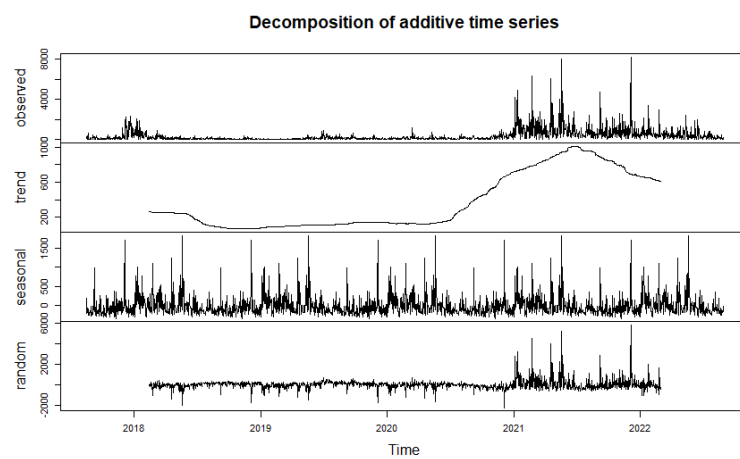


Figure 32 The decomposition plot of lower shadow

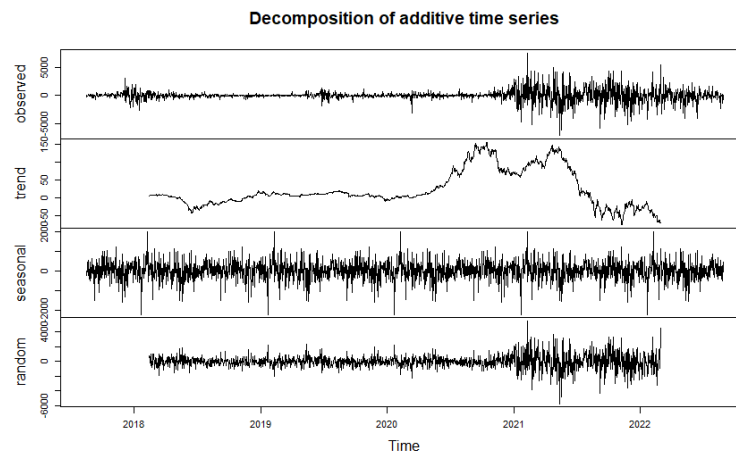


Figure 33 The decomposition plot of upper shadow

Figures 34 to 37 display the time series plot and histogram of the close price, upper shadow, lower shadow, and real body, respectively, after dividing the data into three sets: training set, validation set, and test set. Each figure comprises subfigures that illustrate the time series plot and histogram of each set. Furthermore, Tables 23 to 26 provide a basic statistical summary of each CULR feature after dividing the data.

Table 23 Summary statistics for close price after data splitting

Set	Size (rows)	Percentage (%)	Mean	Std	Min	Max	Skewness	Skewness interpretation
Training set	1472	80	14,344.11	14,146.35	3,189.02	63,649.71	1.15	Highly skewed
Validation set	184	10	49,432.64	8,347.62	35,043.73	67,606.96	0.64	Moderately skewed
Test set	184	10	30,755.42	8,992.05	18,970.79	47,418.50	0.42	Approximately symmetric
Whole data set	1840	100	19,494.10	17,281.56	3,189.02	67,606.96	1.62	Highly skewed

Table 24 Summary statistics for upper shadow after data splitting

Set	Size (rows)	Percentage (%)	Mean	Std	Min	Max	Skewness	Skewness interpretation
Training set	1472	80	232.26	469.61	0.00	10,950.51	0.92	Moderately skewed
Validation set	184	10	628.36	516.10	1.28	2,649.08	0.95	Moderately skewed
Test set	184	10	409.12	324.46	3.10	1,798.30	0.87	Moderately skewed
Whole data set	1840	100	289.56	478.46	0.00	10,950.51	1.07	Highly skewed

Table 25 Summary statistics for lower shadow after data splitting

Set	Size (rows)	Percentage (%)	Mean	Std	Min	Max	Skewness	Skewness interpretation
Training set	1472	80	315.03	593.33	0.00	8,001.14	1.00	Highly skewed
Validation set	184	10	820.84	875.76	0.02	8,246.77	0.90	Moderately skewed
Test set	184	10	397.96	379.84	4.82	2,389.70	0.80	Moderately skewed
Whole data set	1840	100	373.90	628.56	0.00	8,246.77	1.06	Highly skewed

Table 26 Summary statistics for real body after data splitting

Set	Size (rows)	Percentage (%)	Mean	Std	Min	Max	Skewness	Skewness interpretation
Training set	1472	80	28.79	876.25	-7,116.94	7,602.08	0.06	Approximately symmetric
Validation set	184	10	-42.16	1,698.90	-5,819.59	4,325.21	-0.09	Approximately symmetric
Test set	184	10	-89.47	1,147.38	-4,093.20	5,482.90	-0.09	Approximately symmetric
Whole data set	1840	100	9.87	1,016.82	-7,116.94	7,602.08	0.00	Approximately symmetric

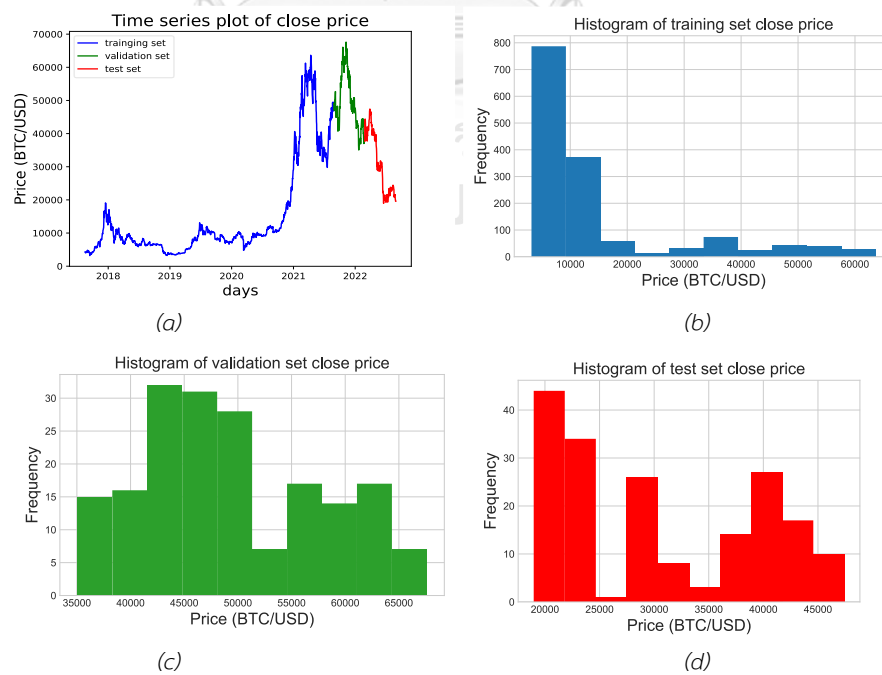


Figure 34 Visualization of close price with data splitting (a) Time series plot (b) Histogram of training set (c) Histogram of validation set (d) Histogram of test set

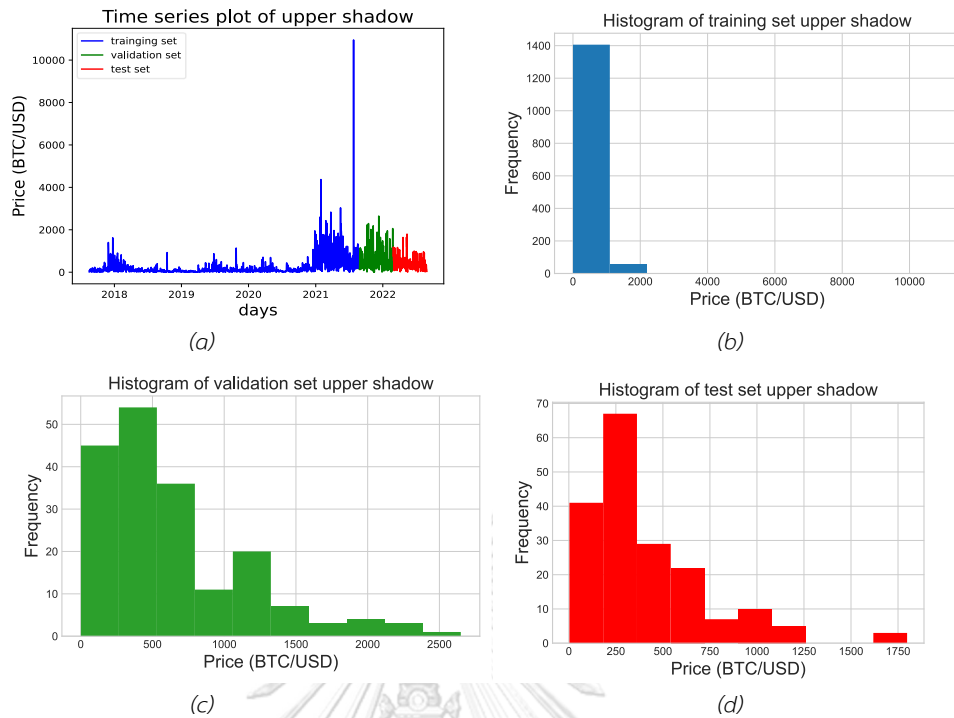


Figure 35 Visualization of upper shadow with data splitting (a) time series plot (b) histogram of training set (c) histogram of validation set (d) histogram of test set

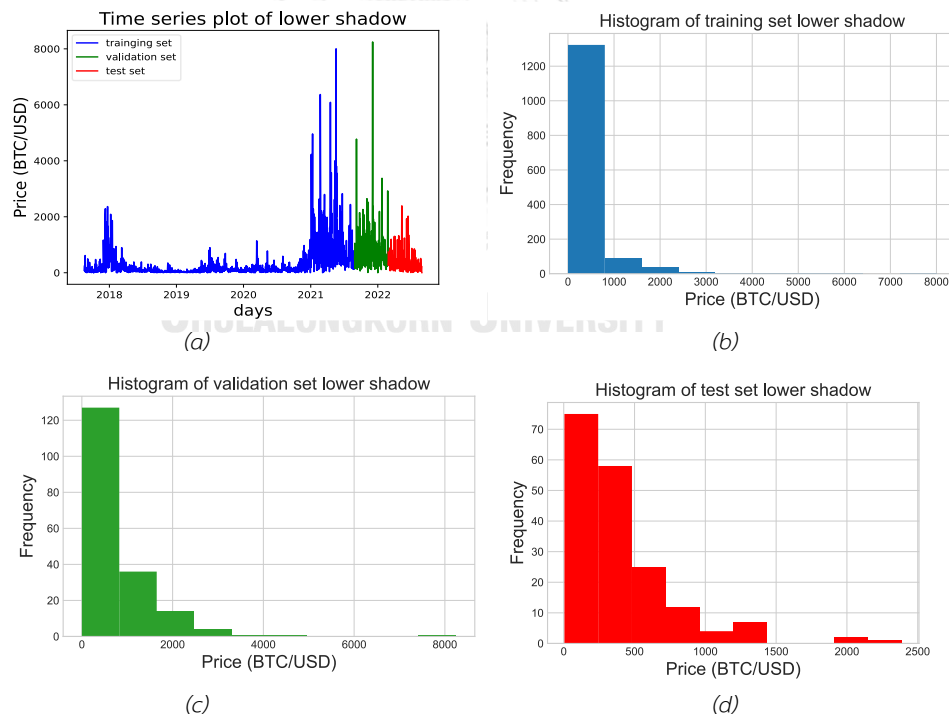


Figure 36 Visualization of lower shadow with data splitting (a) time series plot (b) histogram of training set (c) histogram of validation set (d) histogram of test set



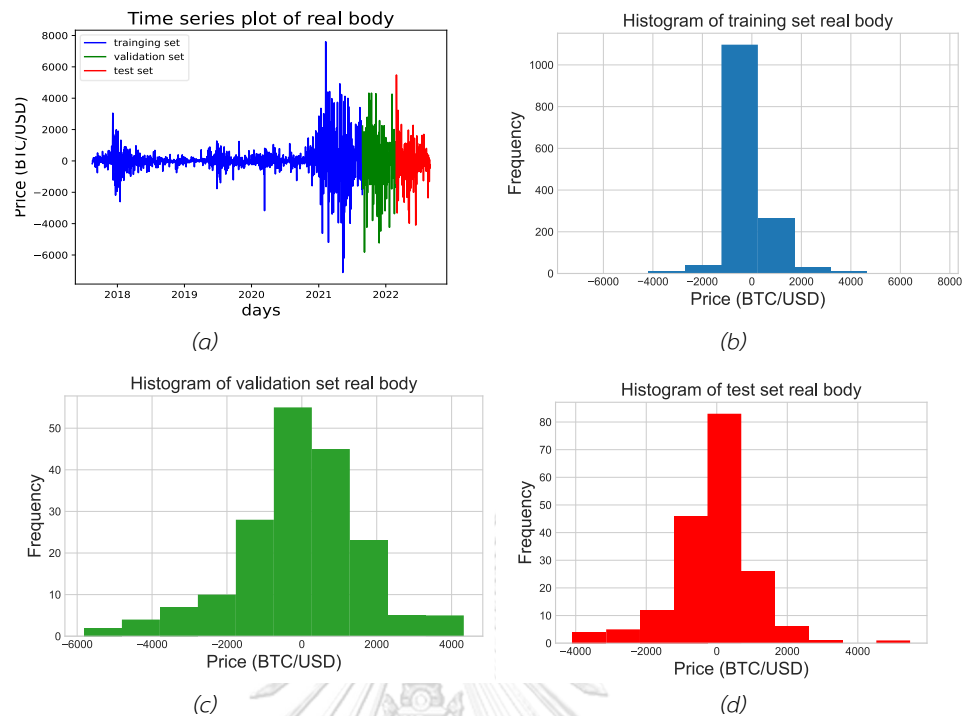


Figure 37 Visualization of real body with data splitting (a) time series plot (b) histogram of training set (c) histogram of validation set (d) histogram of test set

#### 4.5.1.2 Model performance with CULR feature

This section shows the forecasting performance of the LSTM and GRU using CULR feature with various normalization techniques, which were averaged across various sliding window sizes of 3, 5, 7, 15, and 30 days. Note that All the illustrations presented in this section depict the performance of the model after the conversion to OHLC prices. The average and standard deviation of RMSE and MAPE on the test set of LSTM are shown in Tables 27-28 and GRU in Tables 29-30, respectively. Furthermore, Figure 38 illustrates the average RMSE for each price with various normalization techniques and model algorithms, whereas Figure 39 presents the overall average RMSE.

Table 27 shows the performance of LSTM in terms of MAPE, with an average value ranging from 1.58% to 9.42%. The best model performance was achieved using the window z-score technique, with an overall MAPE of 2.40%. Additionally, Table 28 presents the LSTM's performance in terms of RMSE, ranging from 658.55 to 3066.23. The best forecasting error was also achieved using the same technique, with an overall RMSE of 974.90.

Table 29 displays the performance of GRU in terms of MAPE, with an average value ranging from 1.54% to 5.20%. The window z-score technique yielded the best model performance, with an overall MAPE of 2.29%. Furthermore, Table 30 presents LSTM's

performance in terms of RMSE, with values ranging from 626.45 to 2005.53. The best forecasting error was also achieved using the window z-score technique, with an overall RMSE of 925.08.

From Table 27–30, it is obvious that the sliding window normalization techniques yield significantly lower errors than the whole set normalization techniques for both LSTM and GRU. As a result, the sliding window normalization with CULR features also outperforms the whole set normalization same as the model with OHLC features. Furthermore, it appears that LSTM underperforms compared to GRU for the whole set normalization approach. While the sliding window normalization techniques are relatively close, it is not obvious if any technique definitely dominates the others. However, the window z-score technique seems to yield the best forecasting error.

*Table 27 LSTM average and standard deviation of MAPE on the test set*

Normalization Technique	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set z-score	5.44	2.39	5.67	2.26	6.30	2.22	6.47	2.08	5.97	2.10
Whole set min-max	8.45	3.74	8.12	3.65	9.42	4.16	9.29	4.21	8.82	3.66
Window z-score	1.58	0.51	2.24	0.52	2.69	0.56	3.09	0.49	<b>2.40</b>	0.75
Window min-max	1.96	0.49	2.30	0.34	2.81	0.70	2.98	0.38	2.51	0.62

*Table 28 LSTM average and standard deviation of RMSE on the test set*

Normalization Technique	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set z-score	1997.86	806.52	2070.19	820.56	2282.69	671.95	2344.47	675.03	2173.80	701.03
Whole set min-max	2805.03	968.27	2751.58	967.89	3066.23	1068.58	3053.82	1114.17	2919.16	957.92
Window z-score	658.55	195.98	908.03	168.06	1098.59	192.85	1234.42	162.82	<b>974.90</b>	276.96
Window min-max	740.48	137.87	870.37	62.84	1109.18	157.27	1199.61	79.10	979.91	216.11

*Table 29 GRU average and standard deviation of MAPE on the test set*

Normalization Technique	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set z-score	3.84	1.21	4.12	1.25	4.39	0.84	4.62	0.73	4.24	0.99
Whole set min-max	4.74	0.81	4.74	0.82	5.20	0.64	5.15	0.51	4.96	0.68
Window z-score	1.54	0.36	2.07	0.29	2.60	0.32	2.94	0.20	<b>2.29</b>	0.61
Window min-max	2.06	0.53	2.30	0.35	2.75	0.41	2.93	0.22	2.51	0.51

Table 30 GRU average and standard deviation of RMSE on the test set

Normalization Technique	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set z-score	1473.69	444.55	1599.29	518.50	1679.75	315.83	1783.09	318.19	1633.96	392.43
Whole set min-max	1847.68	371.71	1859.99	422.03	2005.53	269.14	2000.61	274.23	1928.45	321.79
Window z-score	626.45	153.33	816.40	108.29	1075.00	99.68	1182.47	55.17	<b>925.08</b>	245.10
Window min-max	799.14	204.39	888.89	130.34	1112.06	92.10	1177.63	39.88	994.43	199.83

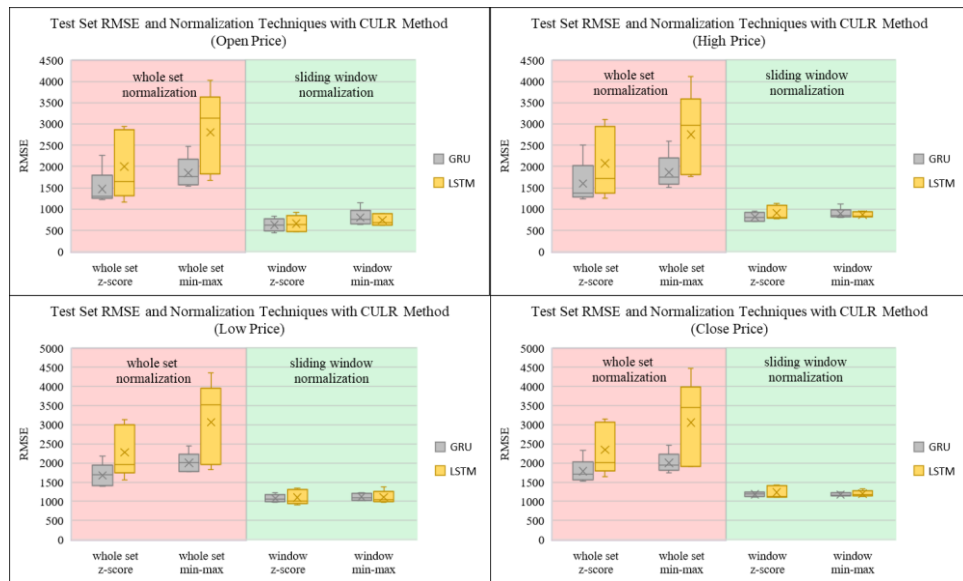


Figure 38 Box plot showing RMSE of each price with various normalization techniques on CULR Method

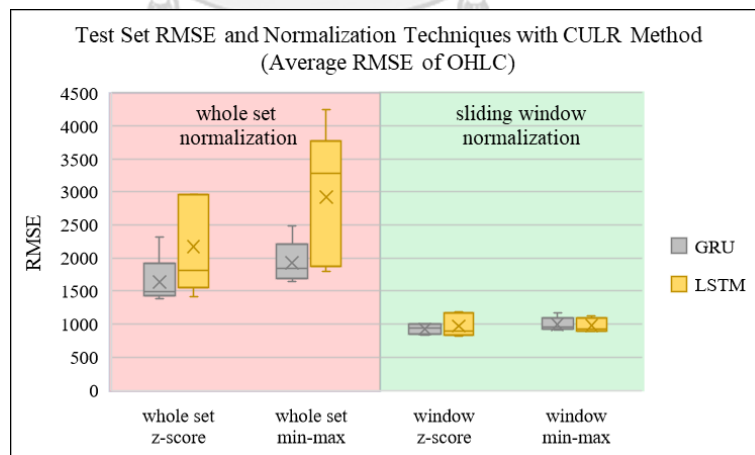


Figure 39 Box plot showing average RMSE of OHLC price with various normalization techniques on CULR Method

#### 4.5.2 Forecasting error comparison

This section presents a comparison of model performance in terms of forecasting error (RMSE), which is averaged across various sliding window sizes of 3, 5, 7, 15, and 30 days, using different features. Figure 40 displays the RMSE for each price of the OHLC method compared to the RMSE of the CULR method with various normalization techniques and algorithms (GRU and LSTM). Figure 41 presents the average RMSE of OHLC prices for both OHLC and CULR methods using the same comparison approach.

Based on Figures 40 and 41, it was found that there is no significant difference in the forecasting error between the OHLC method and the CULR method for the sliding window normalization approach. However, for whole set normalization, the forecasting error varies depending on the normalization technique used. The CULR method performs better for the z-score and relative change techniques, while the OHLC method performs better for the min-max technique. Additionally, the sliding window normalization approach outperforms the whole set normalization approach for both the OHLC method and the CULR method. Among the investigated techniques, the window relative change technique using the OHLC method performed the best in terms of RMSE for both the LSTM and GRU algorithms, based on the average across sliding window sizes. However, the window relative change normalization was not the best when considered for a single sliding window size in our case. The best RMSE when considered for a single sliding window size is shown in section 4.5.4.

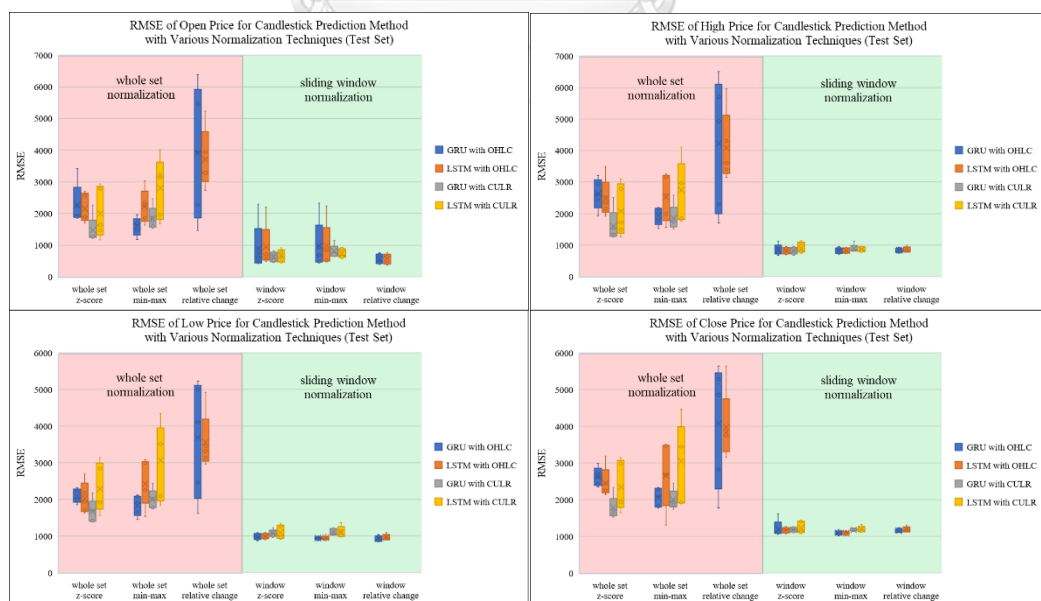


Figure 40 Box plot showing each price RMSE with various normalization techniques for candlestick prediction method performance comparison

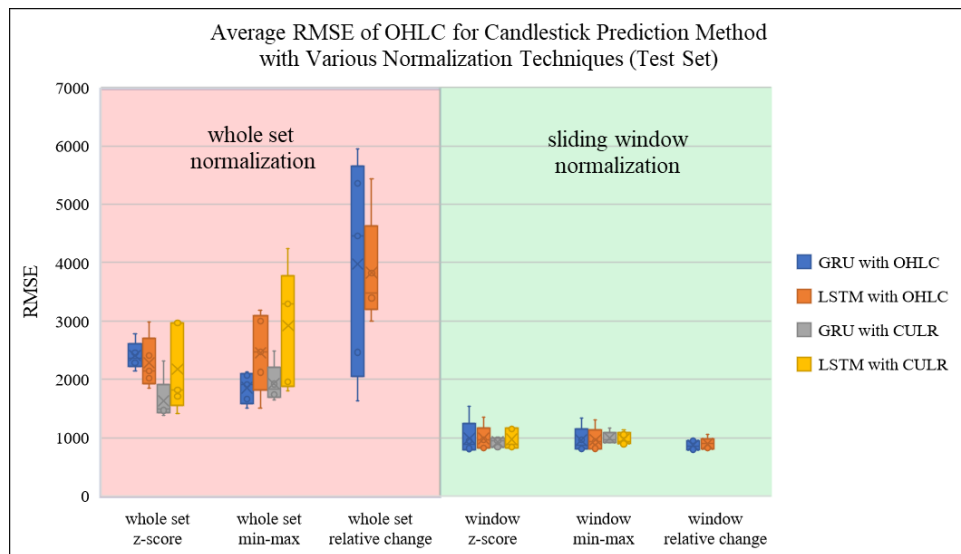


Figure 41 Box plot showing average RMSE of OHLC with various normalization techniques for candlestick price prediction method performance comparison.

### 4.5.3 Accuracy comparison

This section compares the model performance of two candlestick prediction methods in terms of directional prediction. After predicting the OHLC price or converting the CULR prediction to OHLC, the price (number) is converted to price movement (category) or candlestick price direction: bullish (upward) and bearish (downward).

Table 31 displays the average and standard deviation of accuracy on the test set using different candlestick prediction methods and algorithms (LSTM and GRU) average of various sliding window sizes of 3, 5, 7, 15, and 30 days, as well as different normalization techniques. The average accuracy range is 0.45 to 0.55, with the OHLC method achieving the highest average accuracy of 0.55 using the GRU model and sliding window min-max normalization technique. Additionally, Figure 42 illustrates the test set accuracy of each candlestick prediction method with various normalization techniques and algorithms.

Based on Table 31 and Figure 42, it was observed that there is no significantly outperforming method in terms of accuracy. However, the OHLC method shows slightly better performance than the CULR method, except for the whole set min-max normalization technique. Additionally, the sliding window normalization approach appears to be slightly better than the whole set approach. Among the investigated techniques, the window min-max technique using the OHLC Method performed the best in terms of direction accuracy for both LSTM and GRU algorithms.

Table 31 Average and standard deviation of accuracy on the test set

Normalization Technique	OHLC Method				CULR Method			
	GRU		LSTM		GRU		LSTM	
	Average	SD	Average	SD	Average	SD	Average	SD
whole set min-max	0.47	0.04	0.47	0.04	0.48	0.06	0.45	0.01
whole set z-score	0.48	0.04	0.46	0.01	0.45	0.01	0.45	0.01
whole set relative change	0.51	0.05	0.48	0.04	-	-	-	-
Window min-max	<b>0.55</b>	0.04	0.54	0.06	0.49	0.04	0.47	0.03
Window z-score	0.51	0.05	0.50	0.04	0.50	0.05	0.47	0.02
Window relative change	0.50	0.02	0.47	0.03	-	-	-	-

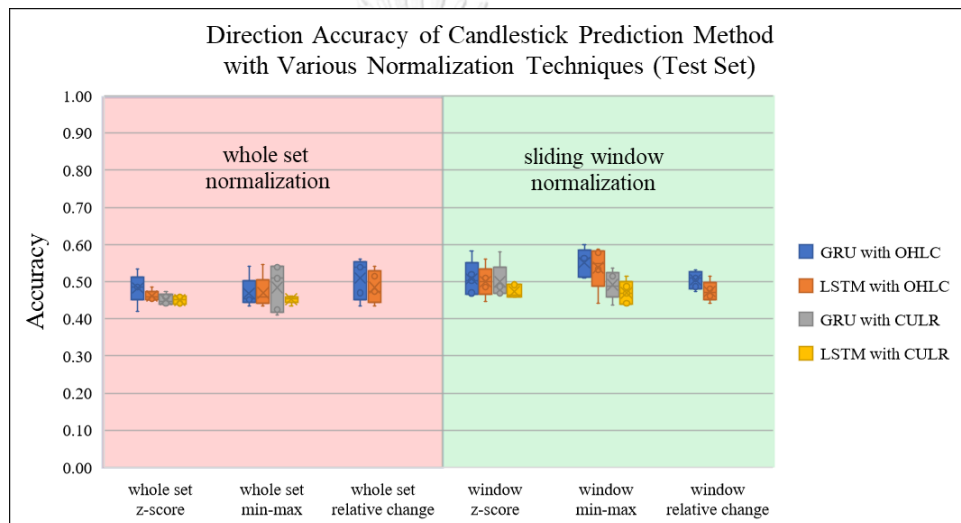


Figure 42 Box plot of test set accuracy with various candlestick prediction method

#### 4.5.4 The best model performance

This section explores the best model performance by analyzing the average RMSE of OHLC prices and direction accuracy of the test set across different sliding window sizes and normalization techniques. Unlike the previous sections, which explored the average across sliding windows to choose the best technique, this section evaluates the performance across various sliding window sizes to identify the best-performing model.

When examining the forecasting error, it was found that the best test average RMSE was 767.71 with an average MAPE of 1.95%. The best test set average RMSE was achieved by the GRU algorithm using the OHLC method with window z-score normalization, and a sliding window size of 15 days. These results are presented in Table 32, which displays the RMSE and MAPE of each predicted OHLC value, as well as the average of the best test set RMSE. Additionally, Table 33

presents the classification metrics of the best test set RMSE, with an accuracy of 50% observed in this model.

*Table 32 RMSE and MAPE of best test set RMSE*

Error	Open	High	Low	Close	Average
RMSE	419.56	680.96	919.07	1051.25	767.71
MAPE (%)	1.06	1.86	2.24	2.65	1.95

*Table 33 Classification metrics of best test set RMSE*

	Precision	Recall	F1-Score	Support
bearish	0.56	0.35	0.43	91
bullish	0.47	0.68	0.56	78
accuracy	0.50			
macro avg	0.52	0.52	0.50	169
weighted avg	0.52	0.50	0.49	169

The best test set accuracy was found to be 60%, achieved by the GRU algorithm using the OHLC method with window min-max normalization and a sliding window of 7 days. Additional classification metrics can be found in Table 35, while Table 34 displays the forecasting errors.

*Table 34 RMSE and MAPE of best test set Accuracy*

Error	Open	High	Low	Close	Average
RMSE	486.30	775.19	915.62	1073.91	812.76
MAPE (%)	1.16	2.04	2.19	2.60	2.00

*Table 35 Classification metrics of best test set Accuracy*

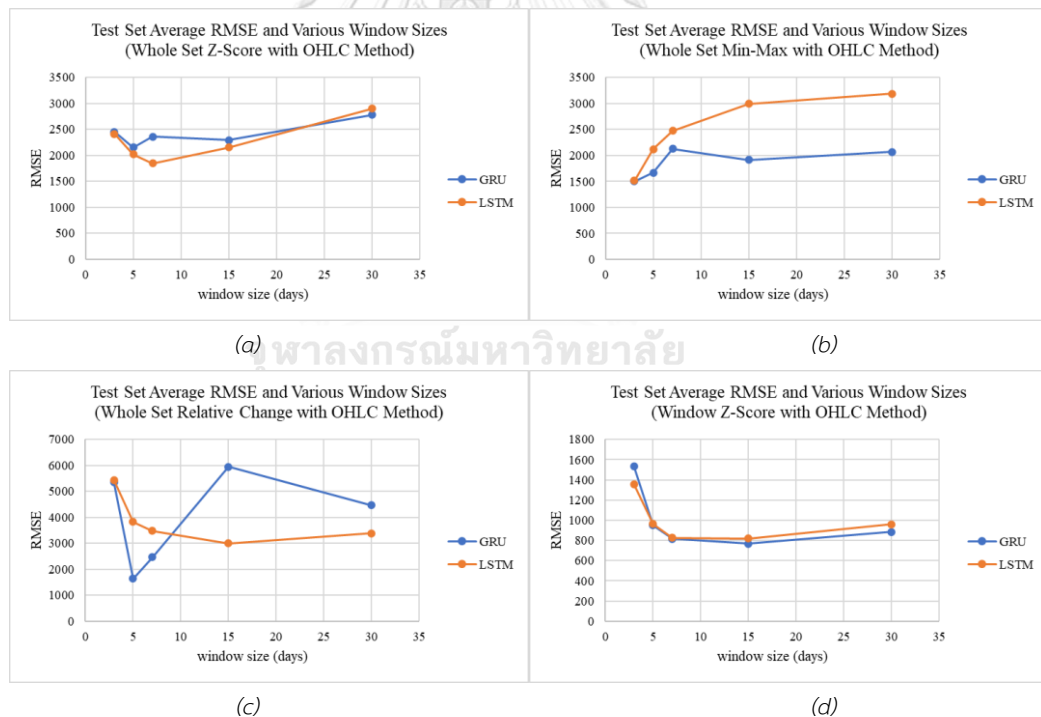
	Precision	Recall	F1-Score	Support
bearish	0.63	0.61	0.62	96
bullish	0.56	0.58	0.57	81
accuracy	0.60			
macro avg	0.60	0.60	0.60	177
weighted avg	0.60	0.60	0.60	177

#### 4.6 Exploring effect of sliding window size

This section explores the effect of sliding window size on forecasting performance, specifically RMSE. We investigated both candlestick prediction methods: OHLC Method and CULR Method, using LSTM and GRU algorithms, as well as various normalization techniques.

Figure 43 displays the effect of sliding window size on RMSE with various normalization techniques for the OHLC method, while Figure 45 displays the same for the CULR method. According to Figures 43 – 45, there are two types of characteristic graphs observed in the investigated techniques: the uptrend graph and the graph with a turning point. It was found that most of the performance curves had the characteristic of a graph with a turning point. However, there was also an unexplainable characteristic graph observed for the performance curve of the whole set relative change with GRU. A summary of these characteristics is presented in Table 36.

According to Table 36, the normalization techniques with a graph displaying a turning point were whole set z-score, window z-score, and window min-max. On the other hand, the techniques with an uptrend graph were whole set min-max (almost) and window relative change.





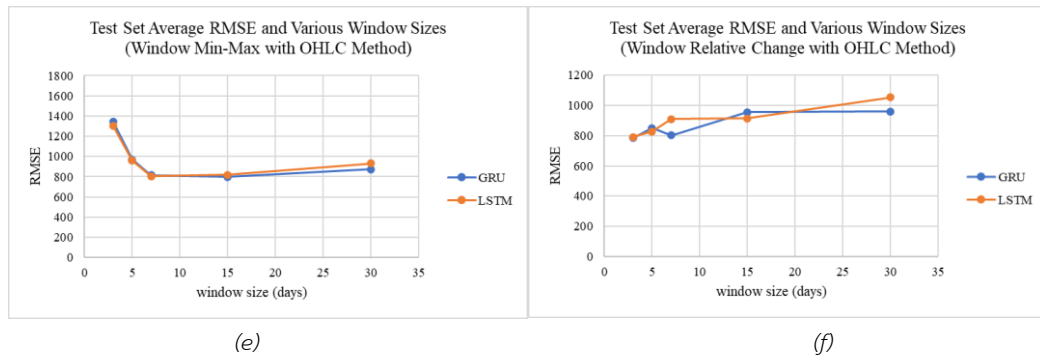


Figure 43 Visualization of the effect of sliding window size on RMSE with various normalization techniques for the OHLC method (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change

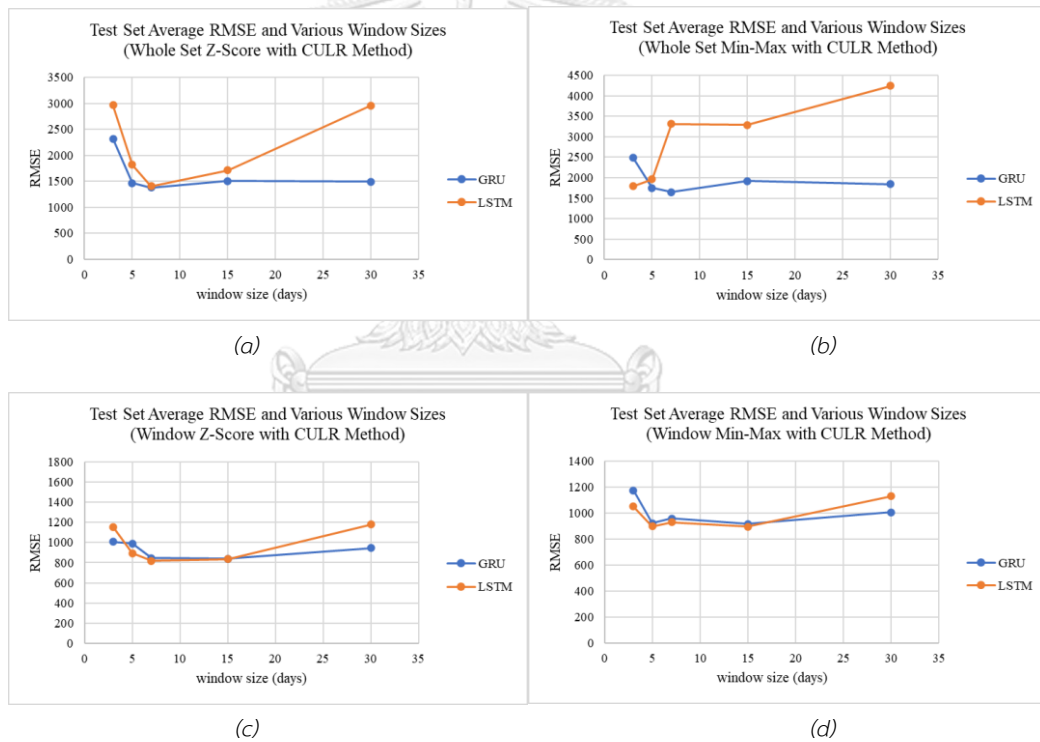


Figure 44 Visualization of the effect of sliding window size on RMSE with various normalization techniques for the CULR method (a) whole set z-score (b) whole set min-max (c) window z-score (d) window min-max

Table 36 Summary of the sliding window effect on the forecasting performance characteristic

Normalization technique	OHLC method				CULR method			
	GRU		LSTM		GRU		LSTM	
	Uptrend	Turning point	Uptrend	Turning point	Uptrend	Turning point	Uptrend	Turning point
Whole set z-score		✓		✓		✓		✓
Whole set min-max	✓		✓			✓	✓	
Whole set relative change	cannot explain			✓	-		-	
Window z-score		✓		✓		✓		✓
Window min-max		✓		✓		✓		✓
Window relative change	✓		✓		-		-	

**Note** ✓ means the technique has that characteristic

#### 4.7 Performance comparison of proposed method with previous research models

This section compares the forecasting performance of our best model with the previous research models. We compared the forecasting performance by the test set normalized root mean square (NRMSE) because most of the research represents the forecasting error by RMSE or MSE, which are different scales since they used different time series ranges. The NRMSE can be calculated by equation 4.1 or by equation 4.2.

$$NRMSE = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}}{\bar{y}} \quad (4.1)$$

$$NRMSE = \frac{RMSE}{\bar{y}} \quad (4.2)$$

We selected the research about cryptocurrency price prediction from chapter 2.2 literature review which predicted the daily price, and our best average RMSE of OHLC. The summary of the comparison is presented in Table 37. Note that for the paper that does not specify split ratio, we assume they used 80:20 for data splitting.

According to Table 37, although our model does not have the best NRMSE performance when compared with all investigated research, it has better performance when compared to models used by other researchers with data ranges similar to our study (no. 5-8).

Table 37 A comparison of our purposed model with the previous research model

#	Authors	Data	Best model	Normalization technique	Split ratio	RMSE	Test set mean	NRMSE (%)
1	(Muniye et al., 2020)	<b>Bitcoin</b> daily price (Jan 2014 to Feb 2018)	- LSTM	min-max (normalized before split)	80/20	0.045 *	6295.51 (0.32)	13.94
2	(Kavitha et al., 2020)	<b>Bitcoin</b> daily price (Jun 2012 to July 2019)	- LSTM	min-max	Does not say (assume 80/20)	95.067	7355.14	1.29
3	(Phaladisailoed & Numnonda, 2018)	<b>Bitcoin</b> daily price (Jan 2012 to Jan 2018)	- GRU	min-max	70/30	0.0044 *	3675.07 (3.07)	0.15
4	(McNally et al., 2018)	<b>Bitcoin</b> daily close price (August 2013 to July 2016)	- RNN	z-score	80/20	5.45%	2637.3	5.45
5	(Hansun et al., 2022)	<b>Bitcoin</b> daily close price (Sept 2014 to Oct 2021)	- GRU	min-max	80/20	1777.31	30421.02	5.84
6	(Tanwar et al., 2021)	Daily price of - <b>Litecoin</b> (Aug 2016 – May 2021) - <b>Zcash</b> (Oct 2016 – May 2021)	- LSTM-GRU hybrid	z-score	80/20	<b>Zcash</b> 0.07* <b>Litecoin</b> 0.14*	<b>Zcash</b> 105.23 (-0.55) <b>Litecoin</b> 125.17 (0.68)	<b>Zcash</b> 12.44 <b>Litecoin</b> 21.02
7	(Vanderbilt et al., 2020)	<b>Bitcoin</b> daily price (Jan 2015 – Apr 2020)	- Gated Recurrent Unit (GRU)	min-max	80/20	403.23	8269.59	4.87
8	(Alkhdhairi et al., 2021)	<b>Bitcoin</b> daily OHLC price (Jan 2017 to Aug 2020)	- Gated Recurrent Unit (GRU) (for historical prediction)	decimal scaling	80/20	0.0037*	8674.95 (0.09)	4.27
9	proposed model	<b>Bitcoin</b> daily price (Aug 2017 to Aug 2022)	- Gated Recurrent Unit (GRU)	window z-score	80/10/10	767.71	30,844.89	2.49

Note \* is RMSE of normalized data, and () is the mean of normalized data

## 4.8 Exploring model performance with before the cryptocurrency boom (before Aug 18, 2020)

To validate the superior performance of the sliding normalization approach over the whole set normalization, we investigated the model's performance in other scenarios by exploring fewer volatile data. This section focused exclusively on forecasting errors and tested the normalization approaches using OHLC price and Bitcoin price data from the Cryptocurrency boom period (before Aug 18, 2020). Note that the experiments in this section also split the data into three sets with the same proportions as when exploring the full range of data.

### 4.8.1 Exploratory data analysis for data before Aug 18, 2020

This section shows the exploratory data analysis for data before August 18, 2020. Figure 45 shows the time series plot of each price. Table 38 shows basic summary statistics for each price. According to Figure 45 and Table 38 all components of the candlestick price, namely the OHLC prices, obviously moved in the same direction during these periods. The finding is also the same as when exploring the full range of data.

After splitting the data into three sets, the data was explored by the time series plot and histogram using open price as representative of OHLC, as shown in Figure 46. In addition, Table 39 shows a basic statistical summary after splitting the data.

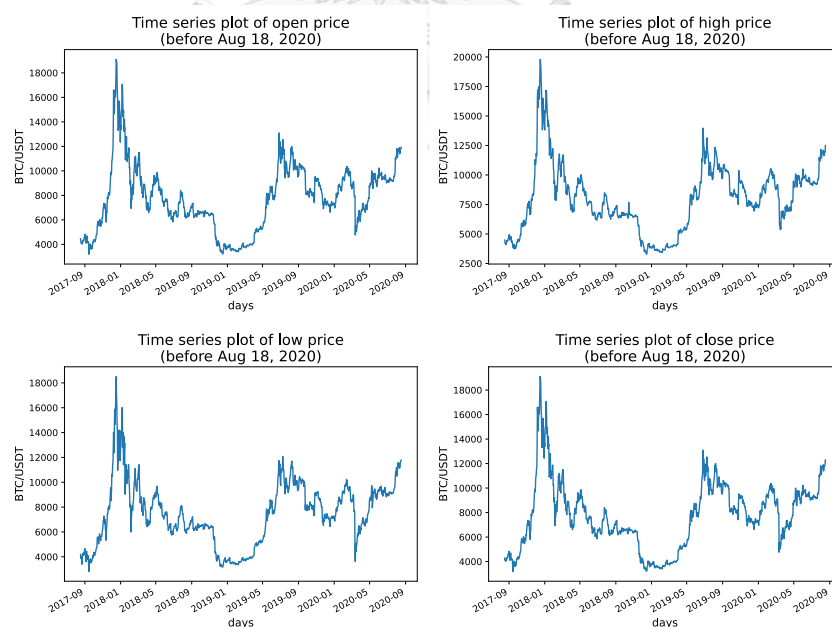


Figure 45 Time series plot of OHLC prices before Aug 18, 2020

Table 38 Summary statistics of each price

Statistics	Open price	High price	Low price	Close price
count	1,098	1,098	1,098	1,098
mean	7,768.81	8,003.09	7,495.66	7,775.89
std	2,728.89	2,854.13	2,542.98	2,730.23
min	3,189.02	3,276.50	2,817.00	3,189.02
max	19,102.66	19,798.68	18,510.00	19,102.66

Table 39 Summary statistics for open price before Aug 18, 2020 after data splitting

Set	Size (rows)	Percentage (%)	Mean	Std	Min	Max	Skewness	Skewness interpretation
Training set	878	80	7,488.70	2,894.52	3,189.02	19,102.66	0.25	Approximately symmetric
Validation set	110	10	8,026.34	1,429.85	4,762.28	10,356.27	-0.35	Approximately symmetric
Test set	110	10	9,747.11	918.35	8,560.73	11,921.32	1.06	Highly skewed
Whole data set	1098	100	7,768.81	2,728.89	3,189.02	19,102.66	0.11	Approximately symmetric

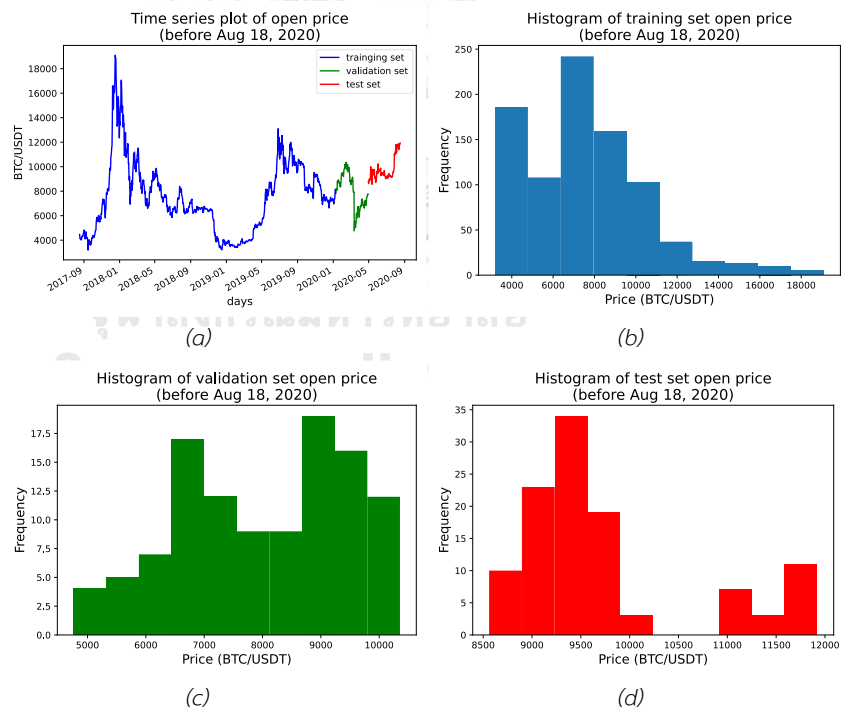


Figure 46 Visualization of open price before Aug 18, 2020 with data splitting (a) time series plot (b) histogram of training set (c) histogram of validation set (d) histogram of test set

#### 4.8.2 Model performance with various normalization techniques

To validate the superior performance of the sliding normalization approach over the whole set normalization, LSTM and GRU models were also trained and tested on the normalized data with various sliding window sizes of 3, 5, 7, 15, 30 days same as when exploring full range of data. The average and standard deviation of MAPE and RMSE for the OHLC prices over various sliding window sizes of LSTM are shown in Table 40-41 and GRU in Table 42-43, respectively. Furthermore, the overall average RMSE of each normalization technique and model algorithm is also illustrated in Figure 47, and each price average RMSE is presented in Figure 48.

From Table 40-43, the finding also same as part 4.3 that all sliding window normalization techniques (the last three rows) yield significantly lower errors than the whole set normalization techniques (the first three rows) for both LSTM and GRU. Moreover, the window relative change normalization yields the lowest average errors in terms of MAPE and RMSE for both LSTM and GRU techniques in most cases. Nonetheless, since the errors of all three sliding window normalization techniques are relatively close, it is not obvious if any sliding window normalization technique definitely dominates the others. However, it can be concluded that the sliding window normalization techniques, in general, significantly outperform the whole set normalization techniques in terms of forecasting performance.

*Table 40 LSTM average and standard deviation of MAPE (%) on the test set using data before Aug 18, 2020.*

Normalization technique	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set z-score	2.52	0.52	2.33	0.12	4.01	0.66	3.15	0.24	3.00	0.79
Whole set min-max	2.86	0.28	2.48	0.28	4.19	1.34	3.13	0.82	3.17	0.99
Whole set relative change	2.53	0.72	2.95	0.47	4.19	0.73	3.37	0.65	3.26	0.87
Window z-score	1.25	0.88	1.71	0.37	1.87	0.27	2.19	0.50	1.75	0.62
Window min-max	1.43	1.02	1.61	0.21	1.98	0.12	2.18	0.25	1.80	0.58
Window relative change	0.97	0.17	1.59	0.12	1.68	0.17	1.90	0.11	1.54	0.38

*Table 41 LSTM average and standard deviation of RMSE on the test set using data before Aug 18, 2020.*

Normalization technique	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set z-score	367.25	53.30	363.89	25.88	514.71	58.51	467.39	33.49	428.31	78.33
Whole set min-max	371.08	34.94	376.86	29.04	484.19	140.37	430.34	86.98	415.62	91.58

Normalization technique	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set relative change	354.65	76.20	435.06	53.89	492.15	96.66	458.29	87.46	435.04	90.07
Window z-score	201.92	142.06	269.88	51.56	267.27	53.24	320.59	63.19	264.91	90.07
Window min-max	215.22	152.55	254.53	22.15	267.53	25.32	311.56	33.28	262.21	81.34
Window relative change	135.86	29.36	256.83	20.38	235.28	32.73	279.80	17.53	226.94	61.08

Table 42 GRU average and standard deviation of MAPE (%) on the test set using data before Aug 18, 2020.

Normalization technique	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set z-score	2.15	0.56	2.36	0.53	3.53	0.87	2.95	0.47	2.75	0.80
Whole set min-max	2.63	0.45	2.92	0.44	4.51	0.91	3.37	0.62	3.36	0.94
Whole set relative change	1.96	0.29	2.32	0.27	3.66	0.51	3.34	0.98	2.82	0.90
Window z-score	1.38	1.13	1.66	0.19	1.81	0.28	2.13	0.47	1.74	0.65
Window min-max	1.53	1.05	1.53	0.16	1.79	0.22	2.17	0.49	1.76	0.61
Window relative change	1.03	0.23	1.61	0.17	1.76	0.39	1.95	0.10	1.59	0.42

Table 43 GRU average and standard deviation of RMSE the test set before data before Aug 18, 2020.

Normalization technique	Open		High		Low		Close		Overall	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Whole set z-score	353.58	101.48	389.12	76.88	473.16	120.91	455.42	83.11	417.82	102.10
Whole set min-max	356.18	49.78	412.22	55.65	519.03	89.48	458.09	65.72	436.38	86.81
Whole set relative change	277.36	31.50	334.54	7.83	433.97	45.59	442.91	97.36	372.20	87.84
Window z-score	219.02	175.92	252.88	16.75	252.47	43.03	307.83	56.48	258.05	93.29
Window min-max	226.53	147.66	237.63	10.55	254.15	35.31	313.95	64.43	258.06	83.35
Window relative change	139.74	35.18	254.67	20.32	241.02	44.05	283.55	17.86	229.74	62.56

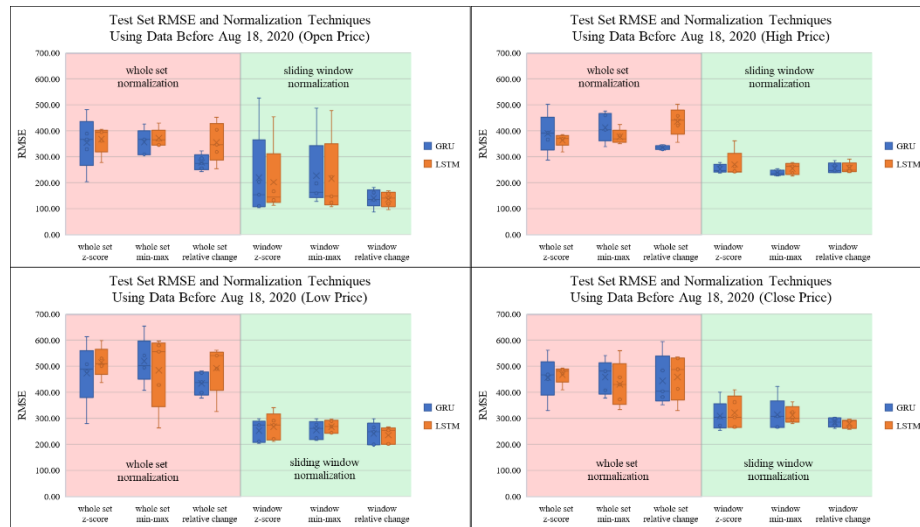


Figure 47 Box plot showing RMSE of each price with various normalization techniques in GRU and LSTM models using data before Aug 18, 2020

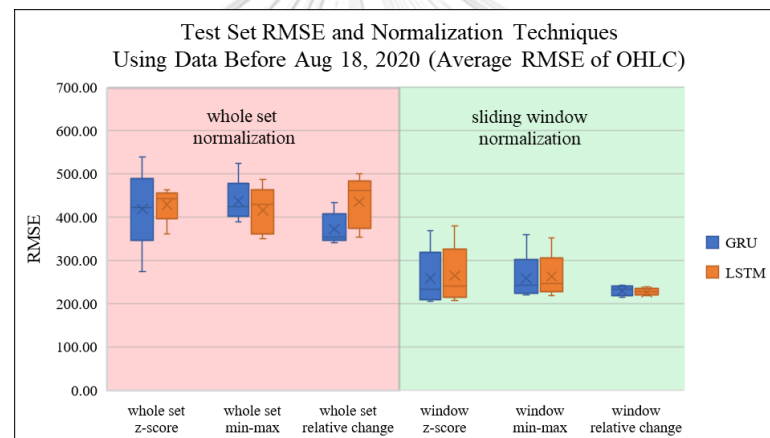


Figure 48 Box plot showing average RMSE of OHLC with various normalization techniques in GRU and LSTM models using data before Aug 18, 2020

### 4.8.3 Impact of normalization on data characteristics

This section explores why sliding window normalization techniques seem to outperform whole set normalization techniques by observing data characteristics, particularly the model input distribution and the sliding window for model input.

#### 4.8.3.1 Impact of normalization on data distribution

This section explores why the sliding window normalization techniques seem to outperform the whole set normalization ones by observing the skewness of data distributions, especially the model input distribution.

Figures 49a-49c show a sample of open-price input distributions of the training, validation, and test sets with a sliding window size of 7 days before normalization. The input



distributions investigated in this section are derived from the sliding window of the model input, which is combined into a single column. Note that the input distribution of each set differs. The distribution of before cryptocurrency boom period is dissimilar to that of the full range data. Specifically, the training set exhibits approximately skewed distribution with a skewness of 0.267, while the validation set and test sets show moderately skewed distributions with skewness values of -0.825 and 0.98, respectively.

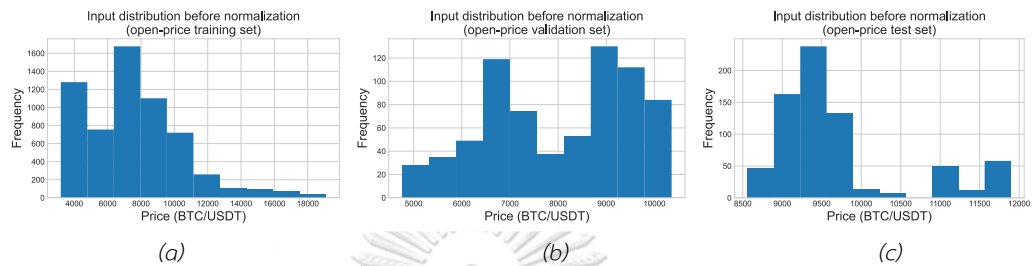


Figure 49 Sample input distribution before normalization of (a) training set (b) validation set (c) test set

Tables 44-46 illustrate the skewness of data distribution after various normalization techniques on the training set, the validation set and the test set, respectively. The skewness values shown in the table are root mean square (RMS) values of five skewness values when the window sizes are 3, 5, 7, 15, 30 days to provide an overall view of skewness across various window sizes. Additionally, sample box plots of open price skewness (magnitude) for the training set, validation set, and test set of input distribution with various normalization techniques are shown in Figures 50-52.

Table 44 Average of training set input distribution skewness

Normalization Technique	Open	High	Low	Close
Without normalization	0.269	0.319	0.206	0.269
Whole set z-score	0.269	0.319	0.206	0.269
Whole set min-max	0.269	0.319	0.206	0.269
Whole set relative change	0.269	0.319	0.206	0.269
Window z-score	0.058	0.154	0.068	0.061
Window min-max	0.048	0.199	0.067	0.052
Window relative change	0.247	0.244	0.227	0.250

Table 45 Average of validation set input distribution skewness

Normalization Technique	Open	High	Low	Close
Without normalization	0.784	0.805	0.916	0.790
Whole set z-score	0.784	0.805	0.916	0.790
Whole set min-max	0.784	0.805	0.916	0.790
Whole set relative change	0.784	0.805	0.916	0.790
Window z-score	0.140	0.202	0.231	0.146

Normalization Technique	Open	High	Low	Close
Window min-max	0.259	0.225	0.257	0.273
Window relative change	0.193	0.201	0.157	0.194

Table 46 Average of test set input distribution skewness

Normalization Technique	Open	High	Low	Close
Without normalization	0.972	1.053	0.850	1.003
Whole set z-score	0.972	1.053	0.850	1.003
Whole set min-max	0.972	1.053	0.850	1.003
Whole set relative change	0.972	1.053	0.850	1.003
Window z-score	0.228	0.270	0.263	0.218
Window min-max	0.280	0.263	0.300	0.280
Window relative change	0.656	0.642	0.607	0.660

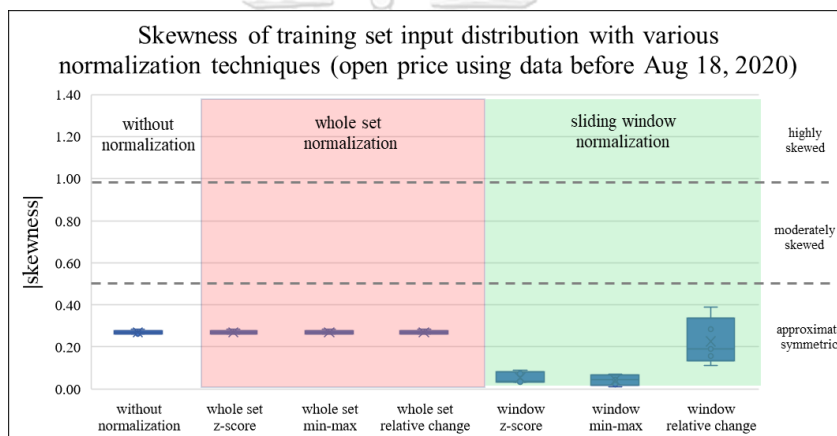


Figure 50 Skewness of training set open price input distribution with various normalization techniques

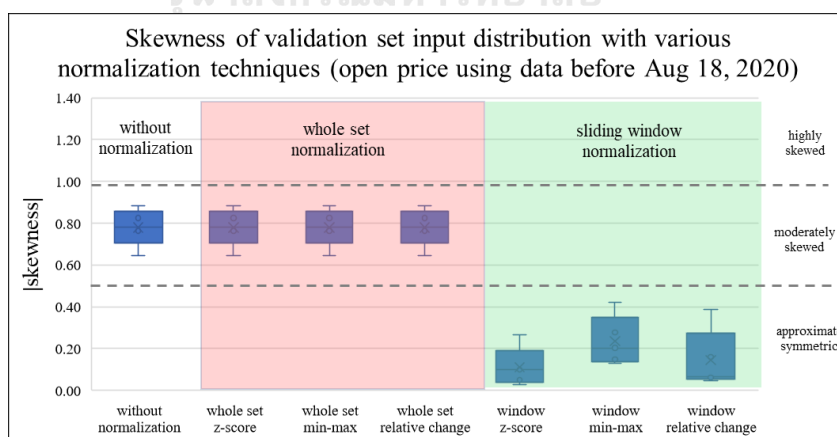


Figure 51 Skewness of validation set open price input distribution with various normalization techniques

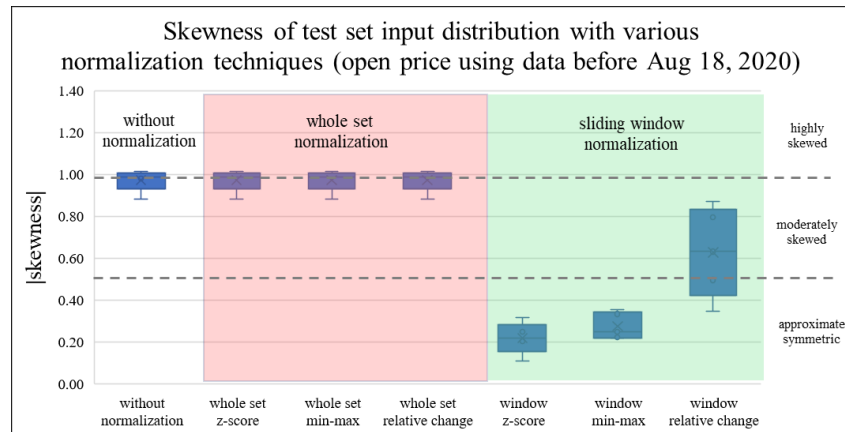


Figure 52 Skewness of test set open price input distribution with various normalization techniques

Table 44 displays the training set skewness before and after each normalization technique. Note that the data before normalization are approximately skewed. Moreover, the whole set normalization techniques do not affect the distribution in terms of skewness at all. On the contrary, the skewness of the data distribution are all lower after the sliding window normalization techniques. In fact, after all three window normalization techniques, the data distributions are close to symmetric because their skewness values are between -0.5 and 0.5. Note that all three sliding window normalization techniques help decrease skewness of highly skewed data. Although the window relative change did not reduce skewness in all cases, the skewness after applying sliding window normalization remained approximately symmetric. Samples of transformed data distributions after various normalization techniques are shown in Figures 53a-53f.

Tables 45 and 46 show similar impact on data distributions after normalization on the validation set and the test set. Again, the whole set normalization techniques do not change the distribution skewness, while the sliding window techniques help reduce the skewness for all distributions to almost symmetric. However, we note that the window relative change normalization technique seems to have less impact in reducing the skewness, especially on the test set, when compared to the window z-score and the window min-max techniques.

The results in this section illustrate that the sliding window normalization techniques generally lead to more symmetric (less skewness) in the input data distribution than the whole set normalization. Potentially due to this impact, more symmetric input distributions then lead to better forecasting performances in both LSTM and GRU models.

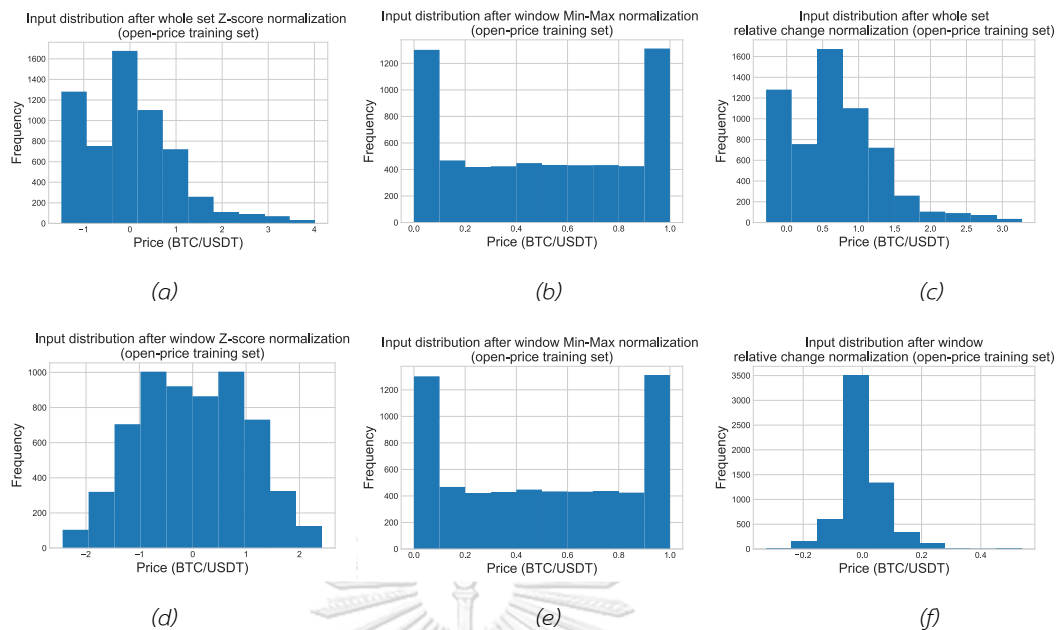


Figure 53 Sample input distribution after normalization by (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change

#### 4.8.3.2 Impact of normalization on sliding window

In order to provide a clearer understanding of why sliding window normalization techniques seem to outperform whole set normalization, we also examined the statistical properties of sliding windows by plotting the mean and standard deviation of the sliding windows over time. Figure 54 displays samples of the sliding window mean for the open price using various normalization techniques with a window size of 7 days. Figure 55 shows samples of the sliding window standard deviation for the open price using the same techniques and window size. Note that sliding window mean and sliding window standard deviation for whole set normalization are equivalent to moving average and moving standard deviation, respectively.

According to Figures 54-55, it was found that sliding window statistical properties of normalized data using sliding window normalization in each sliding window have a stable characteristic, as observed by the value of sliding window mean and sliding window standard deviation that do not significantly change over time. Additionally, the explored statistical properties in each data set have similar values. However, in the case of whole set normalization, sliding window properties vary considerably over time, and the sliding window properties of each data set are significantly different, particularly for sliding window mean.

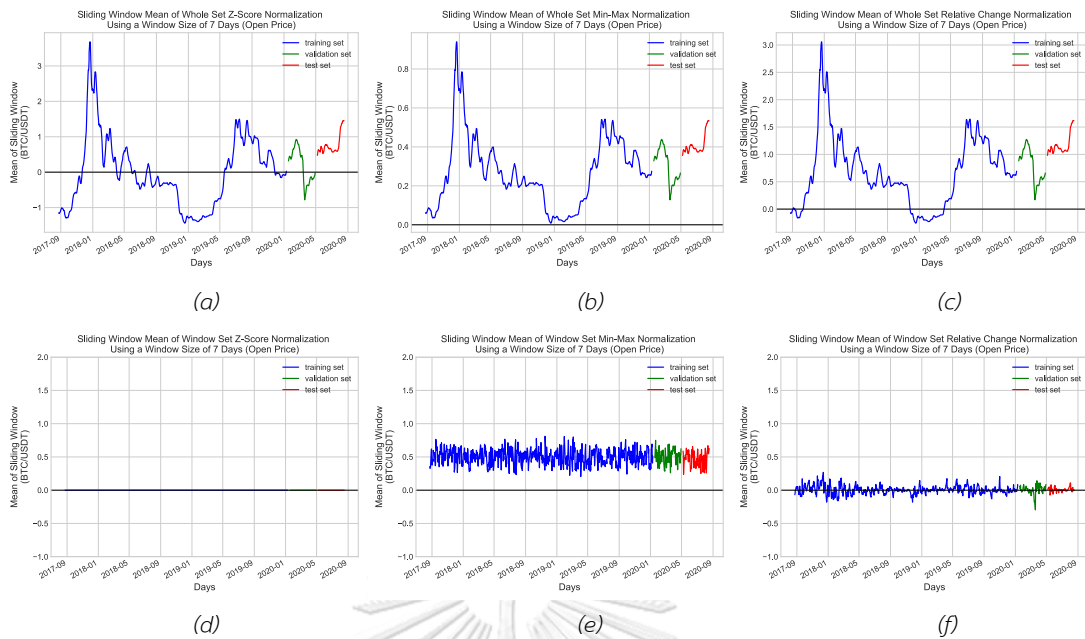


Figure 54 Sliding window mean after normalization by (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change

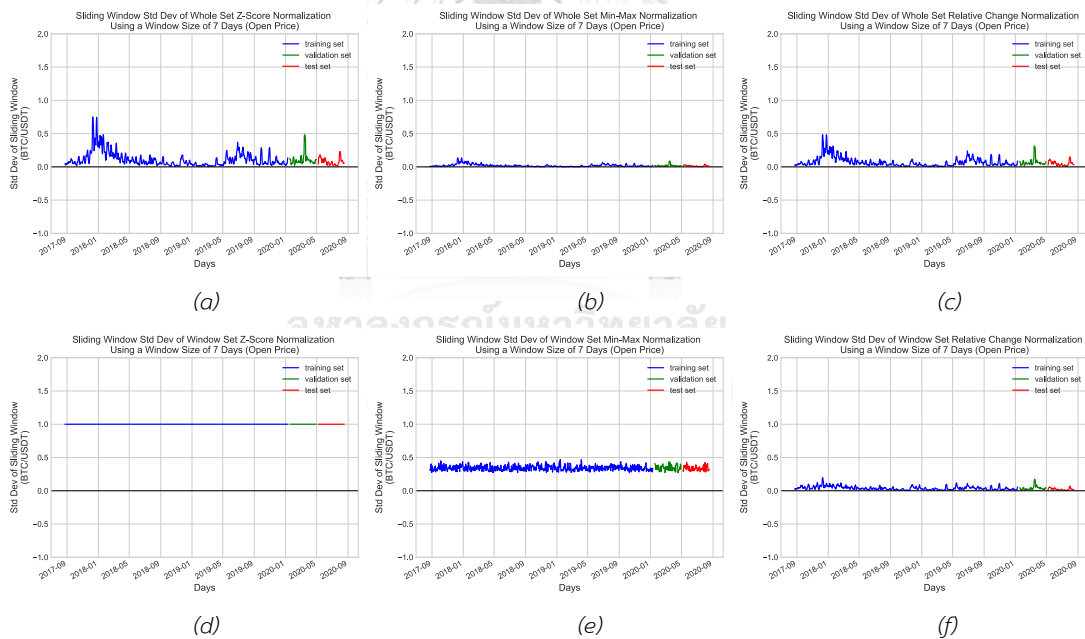


Figure 55 Sliding window standard deviation after normalization by (a) whole set z-score (b) whole set min-max (c) whole set relative change (d) window z-score (e) window min-max (f) window relative change

To further elaborate the findings presented in Figures 54 and 55, we investigated the characteristics of three sliding windows from each data set using all the normalization techniques under consideration. In addition, we used open price as representative of OHLC. Figure 56 presents a time series plot of the open price, along with selected sliding windows for

observation. We selected an example of nine sliding windows including different characteristics (moving upward, moving downward, and moving sideways), with the specified time range for each window provided in Table 47.

Figures 57-60 showcase time series plots of the open price sliding window for the model input using various normalization techniques at each observation window with a sliding window size of 7 days. These include without normalization, z-score normalization, min-max normalization, and relative change normalization. Additionally, Figures 58-60 display each normalization technique with both the whole set normalization approach and the sliding window normalization approach.

Figures 61a-61d present box plots of open price sliding window with various normalization at each observation window. These also include without normalization, z-score normalization, min-max normalization, and relative change normalization. For Figures 61b-61d, each normalization technique is shown with both the whole set normalization approach and the sliding window normalization approach.

According to Figures 58 – 60 and Figures 61b – 61d, we observed that the sliding window of the whole set normalization technique showed a significant difference in the range of each observation sliding window, particularly across the dataset. Conversely, the range of each observation point under the sliding window normalization technique was found to be relatively similar.

The findings of this section indicate that the sliding window normalization techniques have more stable sliding window statistical characteristics. This is evidenced by the sliding window mean, standard deviation, and range of each sliding window used as input for the model. These stable characteristics lead to a similar sequence for prediction, which ultimately results in better forecasting performance in both LSTM and GRU models when using sliding window normalization.

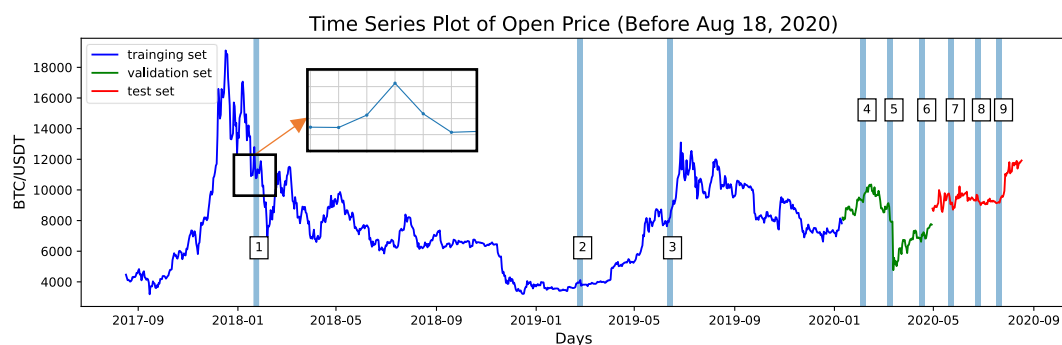


Figure 56 Time series plot of open price with selected sliding windows for observation

Table 47 Time range of the observation windows for data before Aug 18, 2020

Number	Set	Start date	End date
1	Training set	2018-01-20	2018-01-26
2		2019-02-18	2019-02-24
3		2019-06-08	2019-06-14
4	Validation set	2020-02-01	2020-02-07
5		2020-03-05	2020-03-11
6		2020-04-14	2020-04-20
7	Test set	2020-05-19	2020-05-25
8		2020-06-21	2020-06-27
9		2020-07-17	2020-07-23

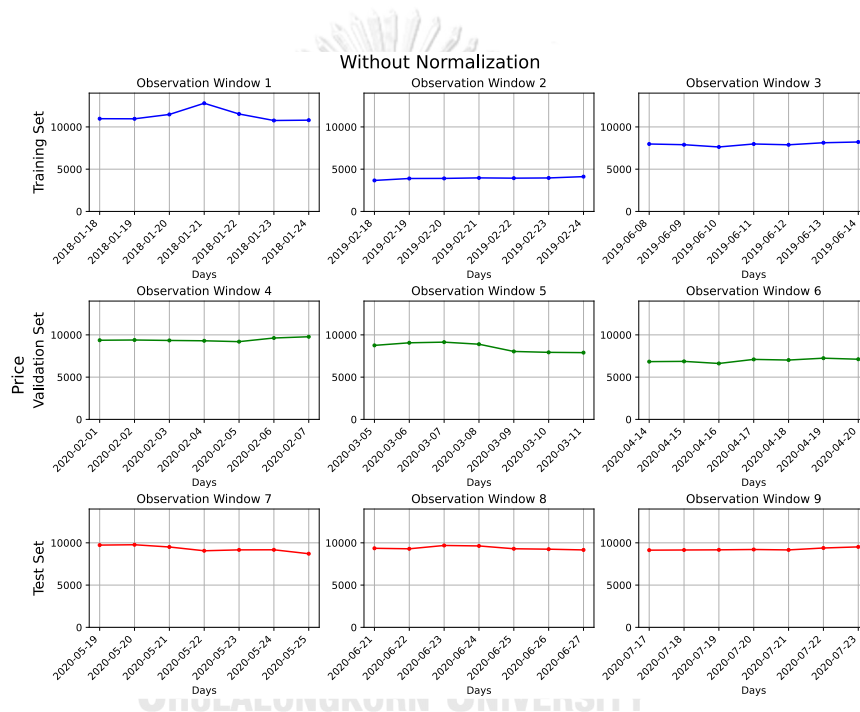


Figure 57 Time series plot of open price sliding window without normalization at each observation window

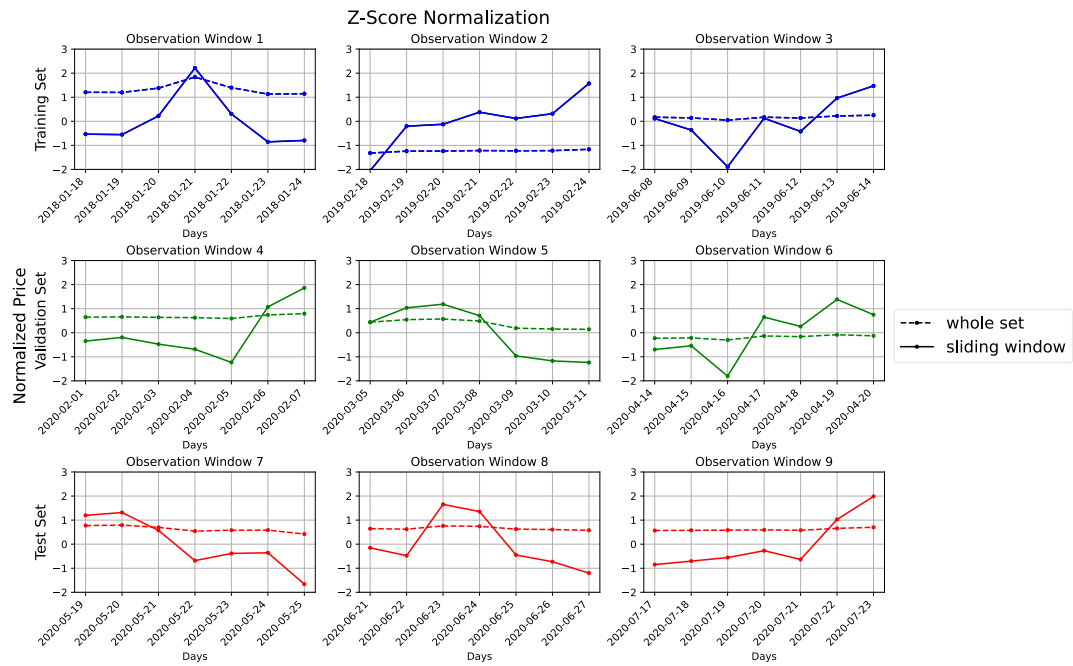


Figure 58 Time series plot of open price sliding window with z-score normalization techniques at each observation window

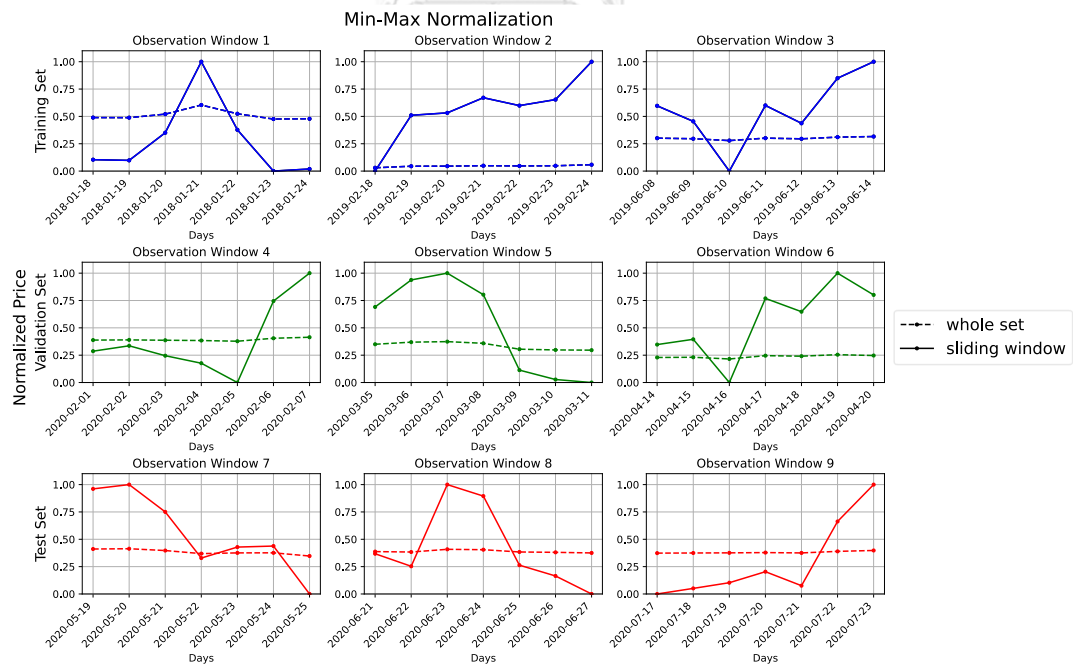


Figure 59 Time series plot of open price sliding window with min-max normalization techniques at each observation window



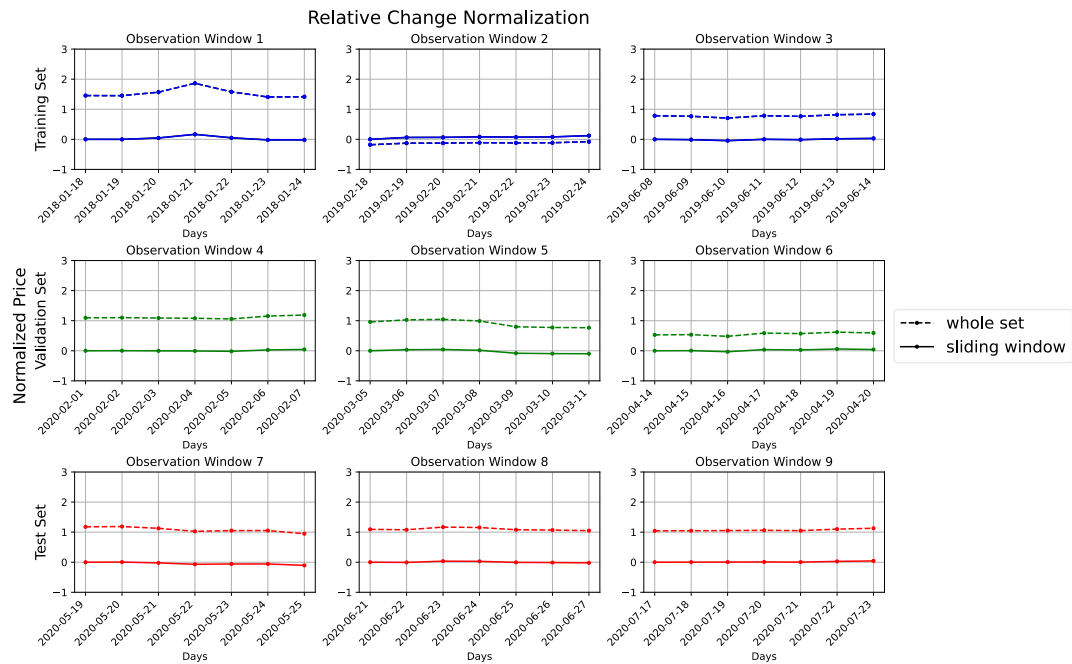


Figure 60 Time series plot of open price sliding window with relative change normalization techniques at each observation window

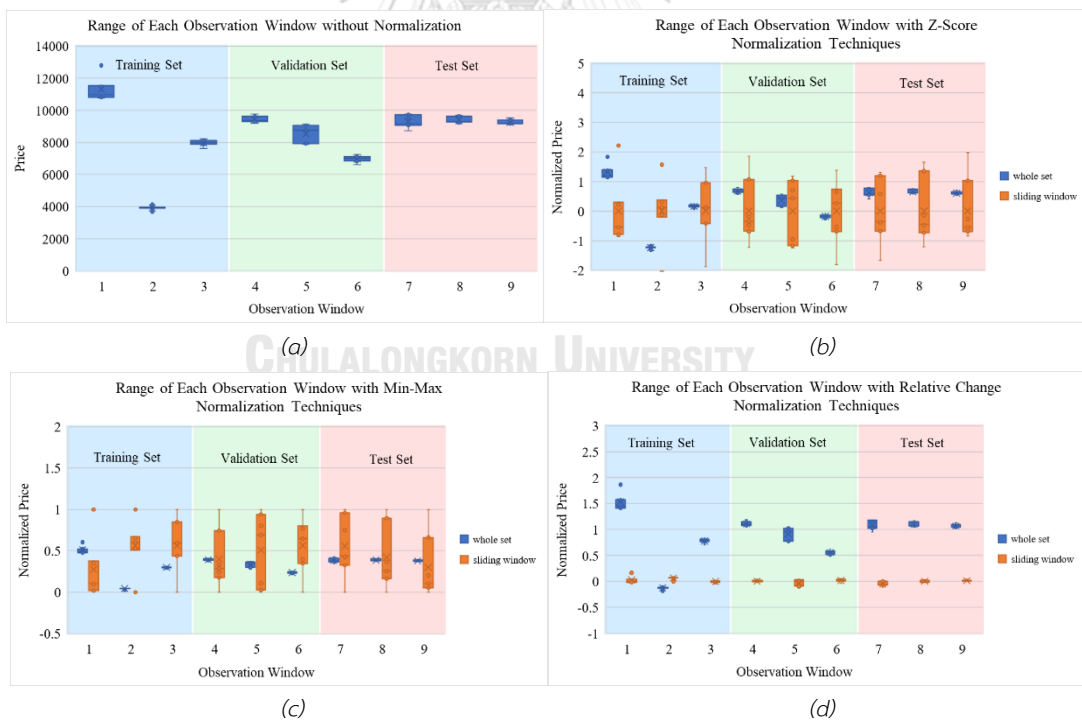


Figure 61 Box plot of open price sliding window with various normalization at each observation window a) without normalization b) z-score normalization c) min-max normalization d) relative change normalization

In summary, our analysis reveals the following findings. Firstly, the sliding window normalization techniques tend to result in a more symmetric input data distribution, as compared to the whole set normalization technique. This increased symmetry potentially explains the improved forecasting performance observed in both the LSTM and GRU models. Secondly, the sliding window normalization approach ensures that the models observe a similar sequence for prediction, which contributes to superior predictive performance when compared to models using the whole set normalization approach.

#### 4.8.3.3 Summary of before the cryptocurrency boom

To summarize the exploration of model performance before the cryptocurrency boom (before Aug 18, 2020), it was found that although the characteristics of this range differ from those of the full range data, the finding that sliding window normalization outperforms whole set normalization on both LSTM and GRU models still holds true, as demonstrated by the RMSE and MAPE. Moreover, the finding that sliding window normalization reduces the skewness of the distribution and enables the model to observe a similar sliding window sequence for prediction also holds true, as previously observed in the full range data

## Chapter 5: Conclusion

The objective of this study is to create a model for predicting the Bitcoin candlestick price of the next day using the recurrent neural network technique, specifically the LSTM and GRU models. Additionally, the study aims to explore feature transformation techniques to enhance the forecasting performance of the neural network, including various data normalization (scaling) methods. Finally, the study aims to investigate additional candlestick price prediction approaches that are not OHLC price prediction, such as candle wick (CULR) prediction.

The best forecasting error was achieved using the GRU algorithm with the OHLC method and window z-score normalization, utilizing a sliding window of 15 days. This resulted in an average RMSE of OHLC of 767.71 or MAPE of 1.95%. Similarly, the best direction accuracy was achieved using the GRU algorithm with the OHLC method and window min-max normalization, utilizing a sliding window of 7 days, resulting in a direction accuracy of 60%.

The main findings are as follows: First, the sliding window normalization generally outperforms the whole set normalization for both LSTM and GRU models in terms of RMSE and MAPE. Second, sliding window normalization techniques generally lead to a more symmetric (less skewed) input data distribution compared to the whole set normalization. This effect potentially leads to better forecasting performance in both LSTM and GRU models. Third, the sliding window normalization approach enables the models to observe a similar sequence for prediction, resulting in better performance compared to models using the whole set normalization. Fourth, there is no significantly outperforming method in terms of both accuracy and forecasting, but the OHLC method seems to be slightly better than the CULR method. Finally, among the investigated normalization techniques, two types of RMSE-sliding window size relationship were observed: uptrend and turning point. Most of the performance curves displayed a turning point characteristic.

Further research could explore alternative normalization techniques and evaluate their performance on different time series datasets. Additionally, more complex RNN models such as Bidirectional RNN, Attention model, and transformer networks, could be explored. Future studies could also explore optimizing the fixed hyperparameters in our study and expanding the search space for the tuned hyperparameters to enhance the model's performance.

## Appendix

*LSTM and GRU test set RMSE with various normalization techniques using OHLC method*

Window size (Days)	Normalization technique	Open		High		Low		Close		Average	
		GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM
3	whole set z-score	2246.56	2239.41	2947.26	2512.25	1999.76	2170.31	2618.14	2710.02	2452.93	2408.00
5		1963.74	1900.64	2436.48	2170.40	1851.90	1711.09	2350.29	2278.33	2150.60	2015.12
7		1860.32	1688.73	2618.67	1924.06	2234.30	1632.04	2725.92	2135.69	2359.80	1845.13
15		1902.49	1880.54	1931.65	2335.96	2329.98	1940.19	2989.77	2444.21	2288.47	2150.22
30		3413.37	2565.15	3203.82	3484.83	2043.20	2337.95	2443.45	3193.98	2775.96	2895.48
3	whole set min-max	1179.72	1308.99	1530.60	1562.39	1457.84	1539.12	1849.10	1637.39	1504.31	1511.97
5		1459.77	1861.84	1780.85	1993.65	1660.28	2258.68	1772.47	2368.24	1668.34	2120.60
7		1955.76	2366.50	2152.32	2583.48	2082.61	2263.06	2332.08	2678.28	2130.69	2472.83
15		1693.59	2244.00	1995.92	3169.05	1899.19	3082.26	2080.10	3488.15	1917.20	2995.87
30		1726.83	3030.66	2166.68	3263.10	2094.42	2991.25	2287.08	3474.10	2068.76	3189.77
3	whole set relative change	5458.66	5237.47	5693.69	5962.00	4999.07	4920.64	5280.13	5649.32	5357.89	5442.36
5		1456.12	3939.93	1707.45	4291.95	1613.62	3316.76	1773.93	3755.31	1637.78	3825.99
7		2283.45	3308.22	2288.98	3610.33	2453.71	3136.63	2830.20	3863.21	2464.09	3479.60
15		6386.56	2738.58	6510.56	3145.68	5232.98	2957.86	5639.28	3161.38	5942.34	3000.87
30		3945.56	3285.91	4928.44	3389.99	4114.48	3448.92	4855.83	3447.01	4461.08	3392.96
3	window z-score	2296.56	2208.98	1122.19	961.37	1101.93	1028.09	1623.00	1235.82	1535.92	1358.56
5		754.36	776.21	894.38	886.94	993.89	951.07	1153.04	1235.85	948.92	962.52
7		469.19	486.94	768.43	753.49	916.32	941.27	1116.49	1127.87	817.61	827.39
15		419.56	574.11	680.96	697.16	919.07	928.39	1051.25	1072.29	767.71	817.99
30		624.69	719.94	761.27	830.54	1030.43	1100.91	1120.43	1189.74	884.20	960.28
3	window min-max	2335.60	2235.99	909.52	880.37	943.39	941.75	1187.97	1159.31	1344.12	1304.36
5		941.27	876.23	908.46	921.71	902.09	909.67	1125.06	1118.99	969.22	956.65
7		486.30	462.28	775.19	740.15	915.62	935.52	1073.91	1072.27	812.76	802.56
15		447.53	532.67	711.35	767.97	942.94	908.05	1085.53	1063.45	796.84	818.03
30		693.61	840.00	771.33	781.69	1007.65	1046.16	1021.80	1057.85	873.60	931.43
3	window relative change	394.81	375.79	762.73	771.16	874.59	894.35	1105.44	1119.27	784.39	790.14
5		500.35	436.26	827.84	810.20	917.22	923.21	1156.72	1142.98	850.53	828.16
7		461.04	644.57	784.21	877.37	850.07	910.63	1117.83	1204.00	803.29	909.14
15		708.42	647.99	900.50	845.15	1007.78	995.26	1200.88	1172.29	954.39	915.17
30		736.45	785.70	895.50	1006.18	1004.12	1102.73	1203.83	1315.78	959.97	1052.60

*LSTM and GRU test set MAPE (%) with various normalization techniques using OHLC method*

Window size (Days)	Normalization technique	Open		High		Low		Close		Average	
		GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM
3	whole set z-score	5.42	7.27	6.99	7.95	4.55	6.61	6.21	8.43	5.79	7.57
5		5.17	5.63	6.59	6.39	4.66	4.41	6.23	6.56	5.66	5.75
7		4.25	4.37	6.13	5.11	5.06	4.05	6.39	5.51	5.46	4.76
15		4.45	4.95	4.42	6.46	5.56	4.86	7.27	6.59	5.43	5.72
30		8.79	8.45	7.96	11.64	4.74	7.35	5.89	10.59	6.85	9.51
3	whole set min-max	2.99	3.32	4.05	4.04	3.52	3.73	4.84	4.13	3.85	3.81
5		3.82	5.01	4.79	5.46	4.07	5.99	4.54	6.45	4.31	5.73
7		5.73	6.93	6.27	7.55	5.78	6.15	6.73	7.76	6.13	7.10
15		4.31	6.02	5.27	9.65	4.65	9.25	5.28	10.95	4.88	8.97
30		4.59	8.86	6.14	9.77	5.61	8.48	6.27	10.29	5.65	9.35
3	whole set relative change	15.94	16.20	15.36	18.22	13.52	15.40	14.37	17.59	14.80	16.85
5		3.43	13.30	4.18	14.34	3.76	11.05	4.32	11.94	3.92	12.66
7		6.27	10.18	5.77	10.64	6.52	9.61	8.69	11.66	6.81	10.52
15		18.15	7.43	18.20	8.75	15.61	7.86	16.41	8.48	17.09	8.13
30		13.03	10.03	15.92	9.86	13.81	10.92	15.87	10.25	14.66	10.27
3	window z-score	4.10	4.03	2.72	2.36	2.72	2.50	3.78	2.97	3.33	2.97
5		1.47	1.39	2.11	2.14	2.46	2.25	2.78	2.88	2.21	2.17

Window size (Days)	Normalization technique	Open		High		Low		Close		Average	
		GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM
7		1.08	1.23	1.92	1.96	2.19	2.32	2.69	2.88	1.97	2.10
15		1.06	1.42	1.86	1.91	2.24	2.32	2.65	2.77	1.95	2.11
30		1.50	1.85	2.10	2.30	2.48	3.12	2.91	3.30	2.25	2.64
3	window min-max	3.94	3.98	2.22	2.09	2.32	2.34	3.06	2.77	2.89	2.80
5		1.73	1.46	2.13	2.25	2.18	2.11	2.64	2.59	2.17	2.10
7		1.16	1.13	2.04	1.91	2.19	2.23	2.60	2.58	2.00	1.96
15		1.12	1.40	1.84	2.08	2.44	2.20	2.64	2.65	2.01	2.08
30		1.71	2.22	2.06	2.19	2.61	2.68	2.78	2.78	2.29	2.47
3	window relative change	0.99	0.91	1.91	1.91	2.13	2.20	2.67	2.69	1.93	1.93
5		1.18	1.06	2.17	2.14	2.12	2.16	2.78	2.73	2.06	2.02
7		1.24	1.59	2.08	2.37	2.23	2.33	2.89	3.11	2.11	2.35
15		1.86	1.67	2.56	2.33	2.60	2.56	3.20	3.04	2.56	2.40
30		2.01	2.14	2.52	2.80	2.75	2.95	3.18	3.44	2.62	2.83

*LSTM and GRU test set RMSE with various normalization techniques using CULR Method*

Window size (Days)	Normalization technique	Open		High		Low		Close		Average	
		GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM
3	whole set z-score	2266.59	2936.19	2504.41	3103.89	2174.29	2844.36	2332.31	2984.35	2319.40	2967.20
5		1310.80	1638.82	1551.95	1717.41	1415.29	1909.20	1602.90	2010.41	1470.24	1818.96
7		1220.91	1165.40	1373.76	1258.04	1392.71	1565.95	1530.74	1645.85	1379.53	1408.81
15		1296.12	1465.83	1322.80	1490.79	1693.07	1956.34	1711.10	1934.07	1505.77	1711.76
30		1274.05	2783.04	1243.52	2780.83	1723.39	3137.61	1738.40	3147.66	1494.84	2962.28
3	whole set min-max	2471.25	1675.70	2588.57	1766.91	2439.54	1830.11	2462.39	1910.88	2490.44	1795.90
5		1592.41	1964.51	1748.65	1861.26	1779.16	2090.86	1865.33	1923.52	1746.39	1960.04
7		1544.89	3231.69	1513.62	3050.55	1781.95	3511.55	1743.97	3448.07	1646.11	3310.46
15		1864.14	3131.91	1803.84	2967.89	2016.87	3543.09	1987.60	3513.23	1918.11	3289.03
30		1765.70	4021.33	1645.26	4111.29	2010.15	4355.56	1943.75	4473.38	1841.21	4240.39
3	window z-score	723.12	778.96	956.14	1133.58	1106.33	1262.26	1241.22	1424.13	1006.70	1149.73
5		828.41	642.15	897.55	791.74	1052.28	1004.90	1184.26	1131.28	990.63	892.52
7		521.25	468.52	728.47	797.31	1019.38	908.70	1114.59	1099.80	845.92	818.58
15		445.68	479.03	795.87	773.99	968.05	971.52	1140.94	1116.79	837.64	835.33
30		613.79	924.11	703.96	1043.51	1228.98	1345.56	1231.34	1400.10	944.52	1178.32
3	window min-max	1150.18	881.50	1120.81	951.75	1185.34	1137.22	1235.21	1228.57	1172.89	1049.76
5		670.55	630.35	843.26	836.17	1015.64	975.47	1150.15	1146.23	919.90	897.06
7		760.76	680.93	833.01	822.40	1085.98	1043.48	1155.56	1172.81	958.83	929.91
15		640.33	613.26	838.88	817.35	1043.50	1019.83	1143.84	1126.97	916.63	894.35
30		773.91	896.34	808.50	924.16	1229.86	1369.89	1203.37	1323.49	1003.91	1128.47

*LSTM and GRU test set MAPE (%) with various normalization techniques using CULR Method*

Window size (Days)	Normalization technique	Open		High		Low		Close		Average	
		GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM
3	whole set z-score	5.99	8.04	6.33	8.19	5.72	7.92	5.89	7.90	5.98	8.01
5		3.26	4.07	3.74	4.44	3.63	4.86	4.12	5.26	3.69	4.66
7		3.20	3.14	3.56	3.51	3.72	4.13	4.19	4.53	3.67	3.83
15		3.55	3.92	3.63	4.16	4.52	5.27	4.41	5.28	4.03	4.66
30		3.22	8.02	3.35	8.06	4.35	9.33	4.50	9.39	3.86	8.70
3	whole set min-max	5.95	4.36	6.07	4.63	6.05	4.93	5.80	5.17	5.97	4.77
5		4.10	4.97	4.39	4.61	4.61	5.39	4.84	4.92	4.48	4.97
7		3.90	10.70	3.87	9.75	4.57	11.82	4.50	11.27	4.21	10.88
15		4.88	9.12	4.77	8.40	5.17	10.45	5.15	10.33	4.99	9.57
30		4.88	13.10	4.58	13.23	5.59	14.51	5.47	14.75	5.13	13.90
3	window z-score	1.81	1.68	2.41	2.55	2.82	2.88	3.10	3.28	2.54	2.60
5		1.92	1.52	2.31	1.97	2.48	2.38	2.92	2.77	2.41	2.16
7		1.14	1.18	1.70	1.79	2.27	2.25	2.62	2.72	1.93	1.99

Window size (Days)	Normalization technique	Open		High		Low		Close		Average	
		GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM
15	window min-max	1.17	1.12	1.98	1.88	2.38	2.33	2.93	2.81	2.11	2.04
30		1.63	2.39	1.97	3.00	3.04	3.59	3.12	3.87	2.44	3.21
3		2.87	2.18	2.88	2.45	2.93	2.81	3.09	2.90	2.94	2.58
5		1.57	1.50	2.07	2.06	2.31	2.24	2.73	2.72	2.17	2.13
7		1.82	1.66	2.04	2.02	2.51	2.39	2.74	2.75	2.28	2.21
15		1.73	1.74	2.16	2.15	2.65	2.62	2.88	2.89	2.35	2.35
30	2.32	2.72	2.37	2.83	3.36	4.00	3.24	3.65	2.82	3.30	

*LSTM and GRU test set direction accuracy with various normalization techniques*

Window size (Days)	Normalization technique	OHLC method		CULR method	
		GRU	LSTM	GRU	LSTM
3	whole set z-score	0.49	0.45	0.45	0.46
5		0.42	0.49	0.44	0.44
7		0.49	0.46	0.46	0.45
15		0.48	0.46	0.47	0.46
30		0.53	0.45	0.44	0.44
3	whole set min-max	0.45	0.55	0.51	0.45
5		0.54	0.46	0.54	0.45
7		0.45	0.45	0.42	0.46
15		0.46	0.46	0.54	0.46
30		0.44	0.44	0.41	0.44
3	whole set relative change	0.55	0.45	-	-
5		0.47	0.54	-	-
7		0.56	0.51	-	-
15		0.54	0.47	-	-
30		0.44	0.44	-	-
3	window z-score	0.47	0.45	0.49	0.46
5		0.52	0.49	0.58	0.46
7		0.58	0.56	0.47	0.46
15		0.50	0.51	0.50	0.49
30		0.46	0.50	0.47	0.49
3	window min-max	0.51	0.44	0.51	0.49
5		0.51	0.55	0.54	0.51
7		0.60	0.59	0.48	0.47
15		0.56	0.53	0.44	0.44
30		0.57	0.58	0.49	0.44
3	window relative change	0.49	0.47	-	-
5		0.51	0.51	-	-
7		0.53	0.48	-	-
15		0.52	0.46	-	-
30		0.47	0.44	-	-

## REFERENCES

- 16 must-know candlestick patterns for a successful trade. (2021). Retrieved 28 June from <https://learn.bybit.com/candlestick/best-candlestick-patterns/>
- Afshine Amidi, S. A. *Recurrent neural networks cheatsheet*. Retrieved 28 June from <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>
- Alkhodhairi, R. K., Aljalhami, S. R., Rusayni, N. K., Alshobaili, J. F., Al-Shargabi, A. A., & Alabdulatif, A. (2021). Bitcoin candlestick prediction with deep neural networks based on real time data. *Computers, Materials, & Continua*, 68(3), 3215-3233.
- Chen, J.-H., & Tsai, Y.-C. (2020). Encoding candlesticks as images for pattern classification using convolutional neural networks. *Financial Innovation*, 6(1), 26. <https://doi.org/10.1186/s40854-020-00187-0>
- Du, J., Liu, Q., Chen, K., & Wang, J. (2019, 15-17 March 2019). Forecasting stock prices in two ways based on lstm neural network. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC),
- Ghorbel, A., & Jeribi, A. (2021). Investigating the relationship between volatilities of cryptocurrencies and other financial assets. *Decisions in Economics and Finance*, 44(2), 817-843. <https://doi.org/10.1007/s10203-020-00312-9>
- Guo, S. J., Hsu, F. C., & Hung, C. C. (2018, 26-28 Dec. 2018). Deep candlestick predictor: A framework toward forecasting the price movement from candlestick charts. 2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP),
- Hansun, S., Wicaksana, A., & Khaliq, A. Q. M. (2022). Multivariate cryptocurrency prediction: Comparative analysis of three recurrent neural networks approaches. *Journal of Big Data*, 9(1), 50. <https://doi.org/10.1186/s40537-022-00601-7>
- Jiang, X. (2020). Bitcoin price prediction based on deep learning methods. *Journal of Mathematical Finance*, 10, 132-139. <https://doi.org/10.4236/jmf.2020.101009>
- Kavitha, H., Sinha, U. K., & Jain, S. S. (2020, 8-10 Jan. 2020). Performance evaluation of machine learning algorithms for bitcoin price prediction. 2020 Fourth International Conference on Inventive Systems and Control (ICISC),
- Kuo Chuen, D. L., Guo, L., & Wang, Y. (2017). Cryptocurrency: A new investment opportunity? *The Journal of Alternative Investments*, 20(3), 16-40. <https://doi.org/10.3905/jai.2018.20.3.016>
- McNally, S., Roche, J., & Caton, S. (2018, 21-23 March 2018). Predicting the price of bitcoin using machine learning. 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP),

- Muniye, T., Rout, M., Mohanty, L., & Satapathy, S. (2020). Bitcoin price prediction and analysis using deep learning models. In (pp. 631-640). [https://doi.org/10.1007/978-981-15-5397-4\\_63](https://doi.org/10.1007/978-981-15-5397-4_63)
- Nakamoto, S., & Bitcoin, A. (2008). A peer-to-peer electronic cash system. *Bitcoin*.–URL: <https://bitcoin.org/bitcoin.pdf>, 4, 2.
- Nayak, S. C., Misra, B. B., & Behera, H. S. (2013). Impact of data normalization on stock index forecasting.
- Panigrahi, S. S., & Behera, H. S. (2013). Effect of normalization techniques on univariate time series forecasting using evolutionary higher order neural network.
- Phaladisailoed, T., & Numnonda, T. (2018, 24-26 July 2018). Machine learning models comparison for bitcoin price prediction. 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE),
- Tanwar, S., Patel, N. P., Patel, S. N., Patel, J. R., Sharma, G., & Davidson, I. E. (2021). Deep learning-based cryptocurrency price prediction scheme with inter-dependent relations. *IEEE Access*, 9, 138633-138646. <https://doi.org/10.1109/ACCESS.2021.3117848>
- Tomar, D., Tomar, P., Bhardwaj, A., & Sinha, G. R. (2022). Deep learning neural network prediction system enhanced with best window size in sliding window algorithm for predicting domestic power consumption in a residential building. *Computational Intelligence and Neuroscience*, 2022, 7216959. <https://doi.org/10.1155/2022/7216959>
- Vanderbilt, D., Xie, K., & Sun, W. (2020). An applied study of rnn models for predicting cryptocurrency prices. *Issues in Information Systems*, 21(2).
- Wang, H., Huang, W., & Wang, S. (2021). Forecasting open-high-low-close data contained in candlestick chart. *Research Papers in Economics*.





จุฬาลงกรณ์มหาวิทยาลัย  
**CHULALONGKORN UNIVERSITY**

## VITA

NAME Sutiwat Simtharakao

DATE OF BIRTH 31 July 1998



จุฬาลงกรณ์มหาวิทยาลัย  
**CHULALONGKORN UNIVERSITY**