

University of Bahrain

College of Science

**Department of
Postgraduate Programs**



Performance Comparison for Deep Learning Models to Predict Top Cryptocurrencies Price

Thesis submitted in partial fulfilment of the requirements for the
degree of M.Sc. In Big Data Science and Analytics

Prepared by:

Mohamed Hamad Khalid Bubashait

ID# 202012424

Supervised by:

Dr. Sawsan Hilal

Assistant Professor

University of Bahrain

Dr. Nabil Hewahi

Professor

University of Bahrain

Kingdom of Bahrain

March 2022



Thesis Discussion Committee Decision

Student Name: MOHAMED HAMAD KHALID BUBASHAIT	Student ID: 202012424
Department: Choose an item.	College: Science
Program: M.Sc. in Big Data Science and Analytics	
Thesis Title: Performance Comparison for Deep Learning Models to Predict Top Cryptocurrencies Price	
Supervisor Name: Dr. Sawsan Hilal + Prof. Nabil Hewahi	Student GPA: 3.39

The Discussion Committee met on 09-May-22 In S41-1002 Sakhir Campus. Upon discussing the student whose details are stated above in public, the discussion committee recommends the following:

The student is awarded a grade A in the thesis subject.

- ☐ Without making any changes to the thesis.
- ☒ After making the modifications attached with this decision.

Discussion Committee Members Approval:

Name

1. Dr. Sawsan Hilal + Prof. Nabil Hewahi
2. Dr. Riadh Ksantini
3. Dr. Manar Abu Talib

Signature

(Advisor)

(Internal Examiner)

(External Examiner)

S. Hilal Nabil Hewahi
[Signature]
[Signature]

Defense Committee's Approval

Acknowledgements

I am thankful to my supervisor, Dr. Sawsan Hilal, for her guidance and support during the process of writing my thesis. Having her support has helped me to better understand scientific research methods and Statistical Learning algorithms that are essential for my research. Furthermore, I am grateful to Professor Nabil Hewahi for introducing me to the concepts of machine learning and deep learning, helping me understand neural network theories, and supporting me in writing my thesis.

Abstract

Today, cryptocurrency investors need accurate predictions to make optimal investment decisions that maximize profits or minimize losses. Moreover, accurate predictions can help policymakers and regulators understand the behaviour of cryptocurrency markets. Cryptocurrency prices, however, are difficult to predict due to their fragility and complexity. Recently, Deep Learning (DL) has been used and presented in academic literature as a potential successor to traditional statistical and machine learning methods for time series problems. DL techniques can efficiently handle time series problems such as nonlinear relationships, multivariate inputs due to their powerful ability to learn from raw data. In this empirical study, the top two current cryptocurrencies, Bitcoin and Ether, are used to compare the performance of 6 DL techniques. These are Fully Connected Neural Network (FCNN), Long-Short Term Memory (LSTM), Bi-directional Long-Short Term Memory (BLSTM), Convolutional Neural Network (CNN), and two other different hybrid architectures constructed using these techniques. The results show that DL techniques effectively predict the prices of cryptocurrencies, with an overall Average Mean Absolute Percentage Error (MAPE) of 9.76, Root Mean Squared Error (RMSE) of 5007.75, and prediction accuracy of 90.24% for Bitcoin. For Ether, the overall averages were a MAPE of 16.41, RMSE of 765.87, and prediction accuracy of 84.13%. The combination of CNN, LSTM followed by FCNN in a hybrid architecture led to the best prediction results for Bitcoin. This combined model outperformed both the basic model and all other models. It achieved a 2.83% lower MAPE, and a 3.03% lower RMSE. On the other hand, CNN's performance on Ether was marginally better than the base model, with a 0.47% lower MAPE and a 0.54% lower RMSE.

Table of Contents

Defense Committee’s Approval	A
Acknowledgements	C
Abstract.....	D
List of Tables.....	G
List of Figures.....	H
List of Abbreviations.....	J
Chapter One: Introduction.....	1
1.1 Cryptocurrency Background Information	2
1.2 Deep Learning	2
1.3 Research Problem.....	5
1.4 Research Questions	7
1.5 Research Objectives	7
1.6 Scope and Limitations	8
1.7 Thesis Organization.....	9
Chapter Two: Literature Review.....	10
2.1 Blockchain Technology and Cryptocurrency Prediction Trends ...	11
2.2 Time Series Forecasting	13
2.3 Deep Learning Techniques.....	14
2.4 Summary of Related Works	18
Chapter Three: Research Methodology.....	21
3.1 Research Data Sources	21
3.2 Research Design	21
3.2.1 Implemented Models.....	25
3.2.2 Evaluation Metrics	33
Chapter Four: Experimentations and Results.....	37
4.1 Experiment Outline	37
4.2 Descriptive Statistics	39

4.3 Base Model FCNN Performance.....	42
4.4 LSTM Model Performance	46
4.5 BLSTM Model Performance.....	50
4.6 CNN Model Performance.....	54
4.7 Combined Model 1 (CNN_LSTM_FCNN)	58
4.8 Combined Model 2 (CNN_BLSTM_FCNN).....	62
4.9 Overall Models Performance Results.....	66
Chapter Five: Discussion, Conclusion and Recommendations	74
5.1 Study Principles Findings.....	75
5.2 Research Limitations and Weaknesses.....	81
5.3 Conclusion	81
5.4 Practical Recommendations	82
References	84
Appendix A: Tools and Libraries.....	
Appendix B: Experiments' Results	
الملخص.....	

List of Tables

Table 3.1 DL Models Architectures.....	27
Table 4.1 Bitcoin Close Price Descriptive Statistics	40
Table 4.2 Ether Close Price Descriptive Statistics.....	41
Table 4.3 FCNN Model Forecasting Performance	44
Table 4.4 LSTM Model Forecasting Performance	47
Table 4.5 BLSTM Model Forecasting Performance.....	51
Table 4.6 CNN Model Forecasting Performance	55
Table 4.7 Combined Model 1 (CNN_LSTM_FCNN) Performance	59
Table 4.8 Combined Model 2 (CNN_BLSTM_FCNN) Performance.....	63
Table 4.9 Models' Average Performance for Bitcoin.....	66
Table 4.10 Detailed Models' Performance for Bitcoin.....	67
Table 4.11 Models' Average Performance for Ether	68
Table 4.12 Detailed Models' Performance for Ether	69

List of Figures

Figure 3.1 Research Methodology	24
Figure 3.2 Selected Deep Learning Architectures	26
Figure 3.3 FCNN Model Architectures	28
Figure 3.4 LSTM Architectures	29
Figure 3.5 BLSTM General Architectures (source: Google Images)	30
Figure 3.6 Convolutional 1D model Architectures	30
Figure 3.7 Hybrid CNN_LSTM_FCNN Model Architectures	32
Figure 3.8 Hybrid CNN_BLSTM_FCNN Model Architectures	33
Figure 4.1 Experiment Pseudo Algorithm	38
Figure 4.2 Bitcoin Close Price Frequency Distribution	40
Figure 4.3 Bitcoin Close Price Time Series	40
Figure 4.4 Ether Close Price Frequency Distribution	41
Figure 4.5 Ether Close Price Time Series	42
Figure 4.6 FCNN Model Fitting and Validation Results	43
Figure 4.7 FCNN Model Prediction Performance	43
Figure 4.8 FCNN Model Forecasting Performance	44
Figure 4.9 FCNN Model Forecasting Performance Metrics	45
Figure 4.10 LSTM Model Fitting and Validation Results	46
Figure 4.11 LSTM Model Prediction Performance	47
Figure 4.12 LSTM Model Forecasting Performance	48
Figure 4.13 LSTM Model Forecasting Metrics Compared to FCNN	49
Figure 4.14 BLSTM Model Fitting and Validation Performance	50
Figure 4.15 BLSTM Model Prediction Performance	51
Figure 4.16 BLSTM Model Forecasting Performance	52
Figure 4.17 BLSTM Model Forecasting Metrics Compared FCNN	53
Figure 4.18 CNN Model Fitting and Validation Performance	54
Figure 4.19 CNN Model Prediction Performance	55
Figure 4.20 CNN Model Forecasting Performance	56
Figure 4.21 CNN Model Performance Metrics Compared to FCNN	57
Figure 4.22 Combined Model 1 Fitting and Validation Performance	58
Figure 4.23 Combined Model 1 Prediction Performance	59

Figure 4.24 Combined Model 1 Forecasting Performance	60
Figure 4.25 Combined Model 1 Performance Compared to FCNN	61
Figure 4.26 Combined Model 2 Fitting and Validation Performance	62
Figure 4.27 Combined Model 2 Prediction Performance	62
Figure 4.28 Combined Model 2 Forecasting Performance	64
Figure 4.29 Combined Model 2 Performance Compared to FCNN	65
Figure 4.30 Models' Average Performance for Bitcoin	66
Figure 4.31 Models' Average Performance for Ether	68
Figure 4.32 Comparison of All Models' Forecasting MAE	70
Figure 4.33 Comparison of All Models' Forecasting MAPE	71
Figure 4.34 Comparison of All Models' Forecasting RMSE	72
Figure 4.35 Comparison of All Models' Forecasting Accuracy %	73

List of Abbreviations

Abbreviation	Full Name
ANN	Artificial Neural Networks
ARIMA	Autoregressive Integrated Moving Average
BLSTM	Bidirectional Long Short-Term Memory
CCi30	Crypto Currency Index. The CCI30 is a market cap weighted index of the 30 largest crypto currencies.
CNN	Convolutional Neural Network
DL	Deep Learning
ETS	Error, Trend, Seasonal method for time series forecasting
FCNN	Fully Connected Neural Networks
GAN	Generative Adversarial Network
LSTM	Long Short-term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
RNN	Recurrent Neural Network
SARIMA	Seasonal Autoregressive Integrated Moving Average
TCN	Temporal Convolutional Network
TSA	Time Series Analysis
TSF	Time Series Forecasting
MAPE	Average Mean Absolute Percentage Error
RMSE	Root Mean Squared Error

Chapter One:

Introduction

Cryptocurrencies have been publicly advocated as a new digital currency exchange mechanism with significant impacts on the global economy. Nowadays, predicting the price of cryptocurrencies is one of the most difficult problems, as there are many unstable factors to consider. These factors contribute to the serious volatility of cryptocurrency prices, which leads to complicated time dependencies (Ioannis E Livieris et al., 2021). Time series forecasting is used in various fields such as economics, weather, transport, environment, sales and production, finance, sports, and healthcare, where future forecasts are beneficial. Forecasts help predict future values based on current and past data. This requires processing a large amount of data to analyse the trend and identify the behaviour of historical data to predict the future (Brownlee, 2017).

In recent years, deep neural networks have been successfully used to predict cryptocurrency prices. They have proven to be a promising solution. Without preprocessing features, deep neural networks can infer temporal dependencies in time series data, however, there have been few empirical studies on cryptocurrencies' time series forecasting problems (Qiu et al., 2017). In order to select the most accurate deep neural network architecture with optimal hyperparameter configuration, empirical comparative studies are required, however, they are extremely complex tasks requiring many years of expertise. Using the current top two cryptocurrencies, Bitcoin and Ether, this thesis seeks to compare various deep learning techniques, identify more effective models for each cryptocurrency, and compare their performance over different forecasting horizons.

This chapter serves as an introduction to the study. Specifically, it gives the background needed to understand the research context; it clearly states the research problem and intended objectives; it highlights the significance of the study and points to study limitations.

1.1 Cryptocurrency Background Information

Cryptocurrency is a currency issued peer-to-peer without a central bank and can be used via a decentralized computer network. It is based on the principle of cryptography and involves the user in the process of issuing and settling transactions. In the first years of their existence, cryptocurrencies gradually gained the attention of the media and the public. With the rapid rise of Bitcoin since April 2013, interest in the currency has grown rapidly. From 2014 onwards, the second generation of cryptosystems emerged, such as Monero, Ether, and Nxt, with new features such as smart contracts, the use of blockchain, or backing with physical assets such as gold (Mougayar & Buterin, 2016). Crypto coins are regarded as one of the most popular and promising types of investments. Nevertheless, they carry the risks of significant volatility and high price fluctuations over time (Ioannis E Livieris et al., 2021).

1.2 Deep Learning

DL is a special subfield of machine learning. It is a multi-stage method for learning data representations using successive layers of neural networks. The number of layers used to construct a model is called the depth of the model. Through a series of simple data transformations (layers), deep learning maps inputs to outputs. The data transformation is learned by example. This transformation performed by a layer is

parameterized by its weights. When learning is referred to in this context, it refers to finding an optimum set of weights for each layer in a network so that the network maps the example inputs correctly to the corresponding targets. To find the best mapping function, a loss function is used to calculate the loss value, which measures how well the network's predictions match the targets. The result of the loss function is used to define the feedback signal used for learning. The optimizer implementing the backpropagation algorithm uses this loss value to update the weights of the network (Goodfellow et al., 2016).

Modern DL often involves tens or hundreds of successive representation layers, learning automatically from training data. In contrast, other machine learning approaches focus on learning only one or two representation layers of the data, so they are sometimes referred to as surface learning (Chollet, 2017). State-of-the-art DL techniques have significantly improved performance in predicting complex time series data such as cryptocurrencies (Sezer et al., 2020). Traditional approaches are unsuitable for processing enormous amounts of data, producing inefficient prediction models (Vijayalakshmi et al., 2021).

In recent years, DL techniques have become a popular approach to forecasting time series. Unlike traditional statistical models that can only model linear relationships in the data, deep neural networks have shown the potential to model complex nonlinear interactions between features. The increase in computational capacity has enabled the development of deeper models, which has dramatically improved their learning ability compared to shallow learning networks. In addition to their ability to learn directly from the data without preprocessing, they offer noticeable advantages when dealing with limited data time series (Torres et al. 2021).

With the increasing availability of data, more sophisticated DL architectures have been proposed and have produced remarkable results. The following paragraphs provide a brief description of the main DL techniques that are used in this study:

- 1) FCNN and a multilayer perceptron (MLP) are the same thing. These techniques consist of multiple layers of fully connected Feedforward Artificial Neural Networks (ANNs) that use Gradient Descend and Backpropagation algorithms to train the network and generate outputs from inputs (Gulli, 2017).
- 2) Recurrent Neural Networks (RNN) for learning relationships over long periods of time are LSTM networks. An LSTM contains a cell state variable that is passed between cells and processed by operation gates (forget, input, and output gates). The operation gates determine how much of the cell state is passed to the output gate. They have independent weights and biases. Thus, the network learns how much of the past output and current input to keep and how much of its internal state to send to the output (Goodfellow et al., 2016; Xingjian et al., 2015).
- 3) BLSTM is an architecture that allows each neural network to input the sequence information in forward and backward directions to obtain the future and past information. BLSTM is different from LSTM because, in regular LSTM, the input flows only in one direction, either backward or forward. (Singh et al., 2021). Bidirectional LSTM merge two independent LSTMs. The time series dataset is simultaneously fed into one layer in a standard temporal order and another in a reverse temporal order. The outputs

of these networks are concatenated at each time step. The networks in this architecture have both forward and reverse sequence information at each step.

- 4) CNN is often used to model visual data such as image and video data. Today, CNN is designed to have several convolutional layers followed by a pooling layer for nonlinear downsampling. The use of the pooling layer reduces the number of parameters that need to be learned in the next layer and allows a deep architecture to be built by stacking multiple convolutional layers. By stacking multiple convolutional layers, each layer learns more features than the previous layer (Vargas et al., 2017).

1.3 Research Problem

Today, cryptocurrency investors need accurate predictions to make optimal investment decisions that maximize profits or minimize losses. Moreover, accurate predictions can help policymakers and regulators understand the behaviour of cryptocurrency markets. Cryptocurrency prices, however, are difficult to predict due to their fragility and complexity. There are a number of existing studies devoted to specific types of DL or specific types of datasets. E. Pintelas et al. (2020) concluded that detailed empirical studies are crucial to incorporating new prediction algorithms, architectures, and techniques. However, it would be difficult to identify which technique would be the most effective and select the best-optimized model for specific cryptocurrencies without additional empirical experiments that allow for comprehensive practical analysis and comparison of cryptocurrency prediction using deep learning (Khedr et al., 2021).

In this study, the researcher aims to fill this gap by conducting a comprehensive experiment to evaluate and compare the suitability and performance of six DL techniques to predict the prices of the two most used cryptocurrencies: Bitcoin and Ether. More specifically, the current study considered the following DL techniques:

- a) FCNN
- b) LSTM
- c) BLSTM
- d) CNN
- e) A combination of CNN, LSTM followed by FCNN
- f) A combination of CNN, BLSTM followed by FCNN.

The two currently most indispensable cryptocurrencies, Bitcoin and Ether, were used to evaluate these models. To compare how well the six DL techniques performed, an optimized FCNN model was developed to provide a general and reliable benchmark.

1.4 Research Questions

The current study was conducted to answer the following questions:

- 1) How effectively and efficiently DL techniques can predict the prices of Bitcoin and Ether?
- 2) What are the effective combinations of different DL techniques in predicting cryptocurrency prices and how do they compare to simple standalone DL techniques?
- 3) How can FCNN model be used as a strong benchmark for measuring the performance of other DL techniques?
- 4) Which DL model provides the optimal predictive performance and for which cryptocurrency, Bitcoin and Ether?
- 5) How much do the research results differ from the relevant studies?

1.5 Research Objectives

This study contributes to the body of knowledge on DL for predicting cryptocurrencies by conducting an experiment to determine the most appropriate deep learning technique, architectures, and models for predicting the price of Bitcoin and Ether and comparing the results with similar recent work on the subject. The main objectives of this work are:

- 1) Evaluate the effectiveness of DL techniques and approaches for predicting cryptocurrency prices.

- 2) Compare the accuracy of DL techniques for predicting cryptocurrency prices and identify the DL architecture that performs best for each cryptocurrency.
- 3) As few comparative studies of DL techniques have been conducted for cryptocurrencies, the purpose of this research is to enrich the research community's knowledge about cryptocurrencies' prediction.

1.6 Scope and Limitations

This study is subject to several limitations. Nevertheless, the results of this study must be interpreted with caution. Some limitations should be considered, such as:

- 1) The study was limited to a univariate forecasting of the cryptocurrency closing price, whereas a multivariate forecasting would have been more comprehensive, but also more complex. The time series analysis phase was excluded. Furthermore, the simple one-step forecasting method was used rather than the multi-step method, which has been shown to be more dependable.
- 2) The data was limited to the two leading cryptocurrencies, Bitcoin and Ether, for the same period from 01-06-2016 to 31-10-2021 for better comparability, but this represents a potential loss of information for the Bitcoin dataset because Bitcoin trading started in 2012.
- 3) Methodology: This study was limited in the number of DL techniques and types. State-of-the-art techniques such as TCNs,

GANs, or Transformer Models should have been included instead of limiting the study to a few DL categories.

- 4) Generalisation: The main limitation in generalizing these results is the limitation to only two cryptocurrencies and the use of a simple prediction strategy.
- 5) Finding similar studies for comparison is quite difficult. This is because the different studies use different DL techniques, different periods of the cryptocurrency dataset with different time series characteristics, and different forecasting horizons.

1.7 Thesis Organization

Throughout this thesis there is a comprehensive review of existing literature on DL for time series forecasting in Chapter 2, the used methodology to conduct the various experiments. Chapter 3 presents the results obtained and Chapter 4, provide a detailed discussion of the results obtained. Finally, the conclusion and future recommendations are provided in Chapter 5.

Chapter Two:

Literature Review

This chapter serves as a background for the review and analysis of related works carried out in this research area. The adopted strategy adopted in reviewing the related work is structured in a thematic approach. The first theme is to understand the current state of knowledge on cryptocurrency fundamentals. A general understanding of blockchain technology is imperative for understanding the cryptocurrency revolution. Bitcoin and Ether cryptocurrencies are currently ranked at the top of the 2000 plus cryptocurrencies existing today (Kulkarni, 2018; Politis et al., 2021). This review focuses on investigating these two cryptocurrencies, trying to understand the data structure that makes up these currencies and their stock prices dynamics using Yahoo Finance web site (Yahoo Finance, 2021). The second theme is understanding the state-of-the-art tools and techniques in using DL to forecast cryptocurrency prices. The third theme of this chapter focuses on understanding the current state of knowledge on time series forecasting using DL. Time series analysis was excluded because it is a different research problem. The fourth theme is to understand the latest research status in using different forecasting methods such as single-step versus multi-step forecasting and their derivatives (Ben Taieb, 2010).

2.1 Blockchain Technology and Cryptocurrency

Prediction Trends

Many technological experts believe that blockchain technology is the next best thing in their world after the Internet. Like the Internet, it is unnecessary to know how it works to use it. However, it is always better to understand the basics of its revolutionary nature. The blockchain is a series of public information and data, accessible to Internet users, and does not have a central storage location. It is distributed database that is safe from hackers and yet has the power to make a legal monetary transaction without fear of fraud and theft.

Initially, blockchain technology was explicitly devised for digital monetary transactions. In the financial world, bitcoin is the first currency based on blockchain technology. However, this technology has found many applications across other platforms, including social media, digital banking, etc. (Tapscott & Tapscott, 2018). In recent years, cryptocurrency has become one of the most popular and promising investment options. Despite this, their significant price volatility, and fluctuations over a short period of time created a need to develop accurate and dependable forecasting models. These models are considered needed for portfolio management and optimization (Lahmiri & Bekiros, 2019). Due to their recent price surges and drop, Bitcoin and Ether have been under the spotlight by the media and financial investors (Sebastião & Godinho, 2021). Many researchers have investigated several factors and patterns behind the cryptocurrencies' price fluctuations using a variety of machine learning and deep learning methods (Danial, 2019).

2.1.1 Bitcoin Prediction

The following section briefly reviews related work on Bitcoin price prediction using DL techniques. McNally et al. (2018) investigated two classification models. They used the Bitcoin price dataset to evaluate two DL architectures. They used simple RNN for the first model and LSTM for the second one and compared the results with ARIMA, the traditional and widely used time series forecasting model. Based on the previous Bitcoin prices, their classification models aimed to predict if the next Bitcoin price will rise or decline. They concluded that RNNs and LSTMs performed better than ARIMAs. Ji et al. (2019) used the LSTM model with a rolling window in a follow-up study. The results indicated that the LSTM model outperformed the other machine learning forecasting techniques. Hua (2020) compared the ARIMA Time Series Model and the LSTM to forecast the Bitcoin price and concluded that the LSTM could make more efficient predictions. Similarly, Shintate and Pichl (2019) proposed a DL-based random sampling model based on combining LSTM and BLSTM in a unique encoding architecture. It was shown that this random sampling had outperformed LSTM-based models. However, none of the works cited above considered CNN-based prediction models or various combinations of DL models.

2.1.2 Ether Prediction

With the goal to create a universal blockchain-based application platform, Ether was launched in 2015. One of its major applications was enabling smart contracts. A smart contract is a piece of code permanently stored on the blockchain that answers users' requests, and it introduces a brand new language in which all computations can be expressed in smart contracts (Tikhomirov, 2017). The distributed nature of blockchain technology makes the Ether platform secure (Antonopoulos & D, 2018).

Kumar and Rath (2020), Politis, A., Doka, K., & Koziris, N. (2021) and Zoumpekis, T., Houstis, E., & Vavalis, M. (2020) concluded that DL models were suitable techniques to capture cryptocurrency price trends in large datasets. LSTM and MLP models were applied to daily, hourly, and minute-by-minute data. The results showed that neural networks could make a better prediction if there is a minor deviation between two subsequent values in prices, and the LSTM model is more robust and precise for long-term dependency. In conclusion of their studies, LSTM did not vary significantly in performance compared to MLP. Furthermore, in their study, they recommended the use of hybrid DL architectures for predicting Ether prices.

2.2 Time Series Forecasting

Observations recorded over time are called time series. The primary temporal structure of time series data makes solving time series problems more challenging since it requires specific data preprocessing and feature engineering techniques (Lazzeri, 2020). Predicting the future using time series forecasting is a vital research topic in many fields. Researchers have focused their attention in the last few decades on developing models that

can capture the underlying patterns of time series so that they can be extrapolated to the future. Ben Taieb (2010) and Abbasimehr et al. (2021) pointed out, for long-term forecasting, a multi-step ahead strategy is preferable to a single-step ahead approach.

A transformation in time series forecasting techniques has taken place in recent years due to the real-time significant data generation that requires enhanced fully automated data-driven forecasting applications (Faloutsos et al., 2019). As a result of the availability of large, diverse, and rich time series datasets, we have seen a shift in this area (Lim & Zohren, 2021). This has led to the emergence of cryptocurrency trading as an exciting topic for time series forecasting problems due to its unique nature and high volatility, making it different from traditional stock markets (Kulkarni, 2018).

2.3 Deep Learning Techniques

As computing power and data availability have increased, today's forecasting models rely heavily on deep learning techniques (Lim & Zohren, 2021). The variety of time series datasets across many domains has led to the development of numerous DL architectures. Modern deep learning models have the ability to learn temporal dynamics through data-driven learning. RNN and LSTM are more preferred in comparison to the traditional MLP due to the temporal nature of the more advanced algorithms (Brownlee, 2017). Torres et al. (2021) provided an overview of deep-learning techniques applied to time series forecasting to provide a comprehensive framework for different application domains for deep learning. They described a range of architectures used for time series data in the past years. Furthermore, they suggested a special emphasis on some

important practical aspects that can significantly influence the results they reported.

Hua (2020) and Torres et al. (2021) found that DL models performance compared to traditional techniques such as ARIMA, SARIMA, ETS, etc., proved effective and outperformed traditional time series analysis and forecasting methods. However, According to Pintelas et al. (2020), they demonstrated that deep learning algorithms outperformed all the other machine learning algorithms in their experiment, but they expected a significant performance improvement in the DL algorithms. Furthermore, Makridakis et al. (2018) evaluated and compared the performances of traditional and modern machine learning and deep learning methods by using more than 1,000 univariate time series forecasting problems. Their study demonstrated that simpler and more traditional methods, such as exponential and linear smoothing, outperformed sophisticated and complex methods. In contradiction to these studies, Brownlee (2017) pointed out that DL techniques have been proved to be efficient when dealing with big data or exceptionally long time series data preprocessing and preparation. This process has been tedious and cumbersome when using traditional methods. However, DL techniques provided automation for features extraction and learning from historical data.

Recently, DL techniques have been a preferred solution for cryptocurrency analysis and forecasting problems. Again, they showed a significant improvement over traditional and machine learning methods (Derbentsev et al., 2020; Hitam & Ismail, 2018; Lahmiri & Bekiros, 2019; Patel et al., 2020; Thanasis et al., 2020). Despite the success of DL applications in many fields, the current state of knowledge on using DL for forecasting cryptocurrencies prices is still limited to comparing them to

traditional and machine learning methods. More comprehensive and detailed comparative studies of DL techniques with the most recently developed techniques are required to enrich the current state of knowledge on this subject (Derbentsev et al., 2021; Ji et al., 2019; Lara-Benitez et al., 2021).

In Derbentsev et al. (2021) study, several sophisticated prediction models were used to model the short-term dynamics of the three most capitalized cryptocurrencies, Bitcoin, Ether, and Ripple. In addition to the ANN and a random forest, they also tested a binary autoregressive tree (BART) model. Their study showed that all models predicted the dynamics of cryptocurrencies, with the mean absolute percentage error (MAPE) for BART and MLP models averaging 3.5%, and for Random Forest models 5%. Chowdhury et al. (2020) applied advanced machine learning prediction models to cryptocurrency indexes and components to forecast future values. As part of their analysis, their primary objective was to forecast the closing price of the Crypto-Currency Index-30 (CCi30) and nine major cryptocurrencies to assist cryptocurrency investors in making trading decisions. To accomplish their goals, they employed diverse types of machine learning methods, including gradient, boosted trees, ANNs, k-nearest neighbour, and robust ensemble learning. They demonstrated that their models' performance seems better than some ML models and competitive to other more enhanced ML models. Their study results showed that LSTM scored an RMSE of 258.78 compared to 2195.67 for K-NN, 639.45 for ensemble learning, and 32.27 for gradient boosted trees.

Traditionally, RNNs were the most common method of modelling sequence data. Transformer models' architecture has started to limit RNNs use for solving time series problems (Nosratabadi et al., 2020). CNN was

initially used for computer vision problems. However, recently it has been successfully applied to time series problems. Consequently, the hybrid networks of CNN and LSTM have achieved a significant improvement compared to their non-hybrid counterparts. FCNN has been considered a strong benchmark for DL models' performance baselining for different time series classification and forecasting problems. Both TCN's and Transformers' models are excellent research directions. However, transformer models have complex architectures and have not been applied to time series problems yet (Ang et al., 2020).

Learning representations from raw data is the crucial advantage of DL models. Most of these architectures used unique architectural designs exploiting CNN and LSTM layers. Data is filtered through convolutional layers to isolate key features and remove noise. In contrast, LSTM layers capture sequence patterns and long-term and short-term dependencies. In spite of their ability to deal with highly nonlinear time-series problems, advanced DL models were shown to be less effective and dependable for forecasting cryptocurrency. More specifically, Pintelas et al. (2020) presented comprehensive research and highlighted the two main reasons for the difficulties in cryptocurrency forecasting. First, cryptocurrency time-series could be classified as a random walk process, and second, the existence of autocorrelation in the errors. Their research revealed that DL models were not dependable for predicting cryptocurrency prices and suggested more advanced algorithmic approaches that would improve their efficiency and reliability.

To address these limitations of DL models, Livieris et al. (2021) introduced a new method for making a stationary time series that will be suitable for fitting a DL model. Ioannis et al. (2020) used ensemble

strategies, i.e., averaging, bagging, and stacking, to improve DL models' forecasting performance and reliability. Furthermore, they evaluated the performance of a range of ensemble models using CNN and LSTM-based models. The authors found that DL and ensemble learning could be applied efficiently to develop reliable and solid cryptocurrency prediction models, though at a prohibitive cost. Patel et al. (2020) presented a hybrid prediction model for Litecoin and Monero cryptocurrencies. It was based on an RNN architecture with combined LSTM and GRU layers. In their study, they reported that the proposed hybrid model outperforms conventional LSTM networks.

2.4 Summary of Related Works

This section summarizes the main points extracted from the relevant studies in cryptocurrency price forecasting.

- DL has been widely used for stock price forecasting and has recently been applied to cryptocurrency forecasting. When it comes to analysis, mining, and forecasting time series, DL outperforms traditional and machine-learning methods.
- Traditional time series forecasting methods require manual features preparation. It is inefficient because it is costly to acquire the required expertise in the field. On the other hand, DL unsupervised learning models can perform time-series analysis and extract the optimal features to be consumed by the supervised learning model to perform the forecasting part.
- In the cryptocurrency market there is no seasonality, and its volatility is high. The ability of DL models to learn representations from raw input data has made them more efficient and cost-effective than traditional methods.

- DL models development and deployment tools, such as Python, TensorFlow, PyTorch, etc., have become simpler and faster to experiment with for different datasets sizes due to the rich arsenal of tools and libraries made available and easily pluggable in any project.
- Based on the success of DL techniques are becoming a dominant technique for time series analysis and forecasting. Conducting more experimental studies comparing the performance of successful DL techniques, such as RNN and CNN, and their assorted flavors for cryptocurrencies time series prediction over different time horizons has become a trending field of study that deserves to get more attention from the research and development communities.
- In the literature on time series forecasting, CNN-based models have been far less widely used since RNNs have been emphasized more.
- A multi-step ahead prediction method is a better substitute for a single-step ahead prediction for long-term horizons. However, there is a shortage of comprehensive empirical studies on this topic.
- Furthermore, there are conflicting views on whether DL techniques can be far better than traditional and ML techniques for time series forecasting problems or no significant enhancement has been realized. While it is agreed that DL techniques have been successful in predicting cryptocurrency price, there is less consensus over whether DL techniques' performance should be compared to traditional ML or themselves.
- However, DL techniques are not adequately covered despite the growing interest in developing DL models for time series forecasting studies and the importance of applying them to cryptocurrencies forecasting. Moreover, relatively few large-scale studies have examined the use of DL models to forecast cryptocurrency time series.

- This study tests and evaluate various deep learning techniques to determine which is the most suitable to predict the Bitcoin and Ether price. Additionally, this study compares this method to similar methods found in recent studies.

Chapter Three:

Research Methodology

3.1 Research Data Sources

The current study used the top two cryptocurrencies, Bitcoin and Ether, time series datasets from 01-06-2017 to 31-10-2021. Yahoo Finance (2021) was used to download Bitcoin and Ether historical data starting from 01-06-2017 to 31-10-2021. In this study, the latest updated data points were downloaded from the same source as those used to evaluate the constructed model. Libraries such as yFinance and Pandas were used to download the latest cryptocurrency prices. The datasets represent univariate time series of closing prices for the selected cryptocurrencies over the period of interest, which makes them suitable for the standard linear and nonlinear deep learning algorithms required to run the relevant experiments.

3.2 Research Design

Figure 3.1 depicts the research methodology design. Once the datasets were downloaded from Yahoo Finance, all attributes were removed, and only the close price was retained. The Keras library was used to normalize the close price values in $[0,1]$ interval using the MinMax Scaler preprocessing class from the scikit-learn library. A Python function to generate time-series datasets for both cryptocurrencies was developed. Two input arguments are used by this function: dataset and lookback window size. The first input is a Numpy array representing the initial dataset, while the second input is the number of previous time steps to be

used for defining the sliding window size. In this study, the value of lag or the sliding windows is taken from a list of the number of days 7, 14, 21, 28, 35, 42, 49, 56, which can be used as well to represent the number of weeks.

After the time series datasets were prepared, the dataset was divided into two parts. An initial set of observations consisting of 80% were used as a training set to train the model, and a second set consisting of the remaining 20% served as a test set. Six different DL models were trained, validated, and tested on these datasets. Table 3.2 describes the six models' architectures and the corresponding hyperparameters configurations after being determined using Keras GridSearch Library. Different activation functions were used according to the model optimal architecture found. Callbacks function with early-stopping patient parameter set to 10 to regulate and determine the optimal evaluation results for the best model. The best model was saved for using it at the forecasting stage.

Since the simplest forecast strategies do not assume much about the nature of the forecast problem and are easy to implement and calculate, a model can be dependable and improvable if it produces acceptable forecasting results based on a simple forecast strategy. The basic principles of simple forecast strategies are:

- 1) Naive or using observations values directly.
- 2) Average, or using a statistic calculated on previous observations.

In this research, a simple strategy, recursive single-step output forecasting, was used. It is a many-to-one model that takes as input a sequence and produces a scalar rather than the multi-step output (Ben Taieb, 2010). Chapter 4, section 4.1, explains how the different experiments were conducted. A total of ten experiments for six models, two cryptocurrencies, and eight forecasting windows were conducted, which generated 920 records for the entire experiment. Furthermore, there are 160 records for each model consisting of 10 repeated experiments, eight forecast windows, and two cryptocurrencies, Bitcoin and Ether.

In this study, six DL models of different architectures were compared across different forecasting window sizes. Specifically, univariate time series datasets were used to forecast the closing prices of Bitcoin and Ether. The study provides evidence to support other studies that claim that DL techniques predict crypto price correctly, and tests whether different DL techniques can be used to predict crypto price more effectively than basic DL techniques. Performances of DL techniques were measured and compared using MAPE and RMSE.

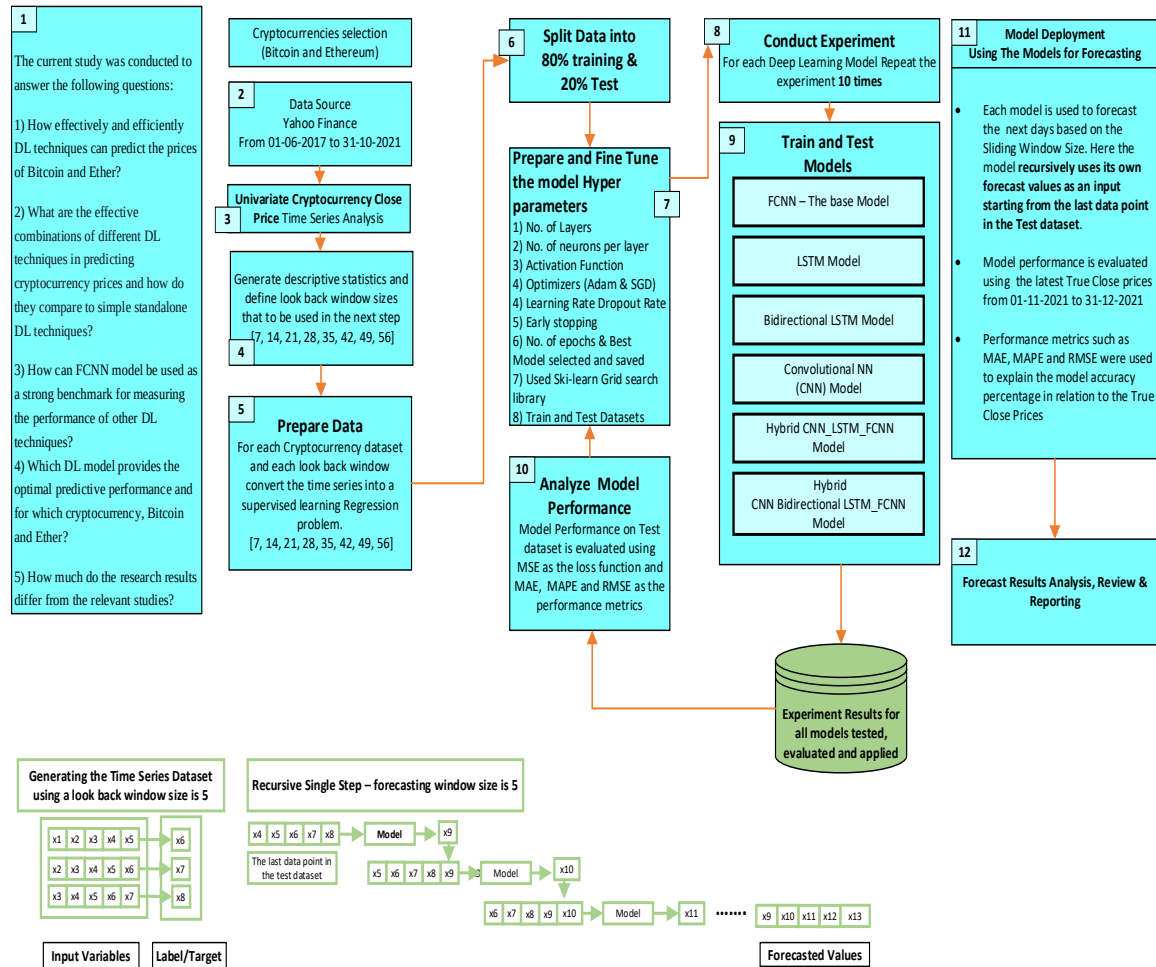


Figure 3.1 Research Methodology

3.2.1 Implemented Models

Figure 3.2 shows the different possible categories of state-of-the-art DL techniques applied to time series forecasting problems. This classification was made on the original purpose of these techniques when they were invented. These techniques provide flexible design architectures depending on the problem being addressed. They vary in sophistication from quite simple such as MLP, or vary complex, depending on the type and complexity level of the tackled problem. However, for this study, only techniques, which are highlighted with green, were selected from each category. A structured experiment was designed to assess and compare the performance of 6 DL techniques. These are:

- a) FCNN
- b) LSTM
- c) BLSTM
- d) CNN
- e) A combination of CNN, LSTM, followed by FCNN
- f) A combination of CNN, BLSTM, followed by FCNN

An optimized FCNN model was designed and used as a baseline to provide a general and reliable benchmark to compare the other five techniques' forecasting accuracy. Six different models for each cryptocurrency dataset have been evaluated, covering a broad range of architectures and training parameters. Finally, the whole experiment was repeated ten times, and average measures were calculated.

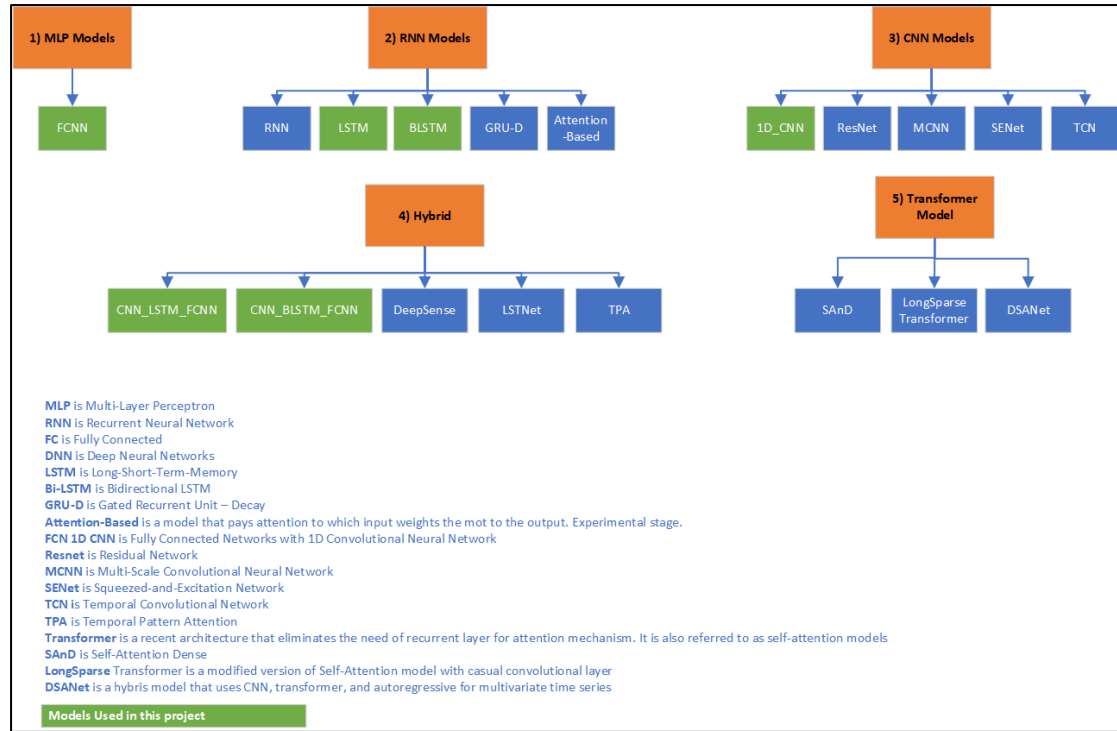


Figure 3.2 Selected Deep Learning Architectures

FCNN technique from the MLP category has been used as the baseline model to assess the performance of other models. LSTM and the BLSTM were selected from the RNN category. From the CNN category, 1-Dimensional CNN was selected. Two more DL techniques were added by combining different DL techniques into two hybrid models. Table 3.1 lists the selected DL techniques and their respective architectures that were designed and used in our research, along with the values of the optimal hyperparameters found for configuring these models.

Table 3.1 DL Models Architectures

Model Name	Type	Model Architecture	Model Description	Dropout %	Learning Rate	Activation Function	Optimizer	Loss Function	Metrics	Early Stopping
FCNN	Sequential	model.add(Dense(50, activation = 'relu', input_shape = (look_back, n_features))) model.add(Dense(50)) model.add(Dense(25)) model.add(Flatten()) model.add(Dense(units=1,activation='linear')) model.compile(loss='mean_squared_error', optimizer=keras.optimizers.Adam(learning_rate=LEARNING_RATE), metrics=['mae', 'mape'])	One input layer of 50 neurons, 3 hidden layers with 50, 50, 25 neurons respectively and one output layer with 1 neurons.	None	0.001	Relu for input layer and for the hidden layers and linear for the output layer	Adam	MSE	MAE, MAPE, RMSE	Yes (patient = 10)
LSTM	Sequential	model.add(LSTM(50, activation='relu', return_sequences=True, stateful = False, input_shape=(look_back, n_features))) model.add(Dropout(DROP_OUT)) model.add(LSTM(128, activation='tanh', return_sequences=True)) model.add(Dropout(DROP_OUT)) model.add(LSTM(64, activation='tanh', return_sequences=False)) model.add(Dropout(DROP_OUT)) model.add(Flatten()) model.add(Dense(units=1,activation='linear')) model.compile(loss='mean_squared_error', optimizer=keras.optimizers.Adam(learning_rate=LEARNING_RATE), metrics=['mae', 'mape'])	One input layer of 50 neurons, 3 hidden layers with 128, 64 and 32 neurons respectively and one output layer with 1 neurons.	0.2	0.001	Relu for input layer, tanh for the LSTM hidden layers and linear for the output layer	Adam	MSE	MAE, MAPE, RMSE	Yes (patient = 10)
BLSTM	Sequential	model.add(Reshape((look_back,1), input_shape=(look_back, n_features))) model.add(Bidirectional(LSTM(50, activation='tanh'))) model.add(Dense(1)) model.compile(loss='mean_squared_error', optimizer=keras.optimizers.Adam(learning_rate=LEARNING_RATE), metrics=['mae', 'mape'])	One input layer of 50 neurons, 3 hidden layers with 128, 64 and 32 neurons respectively and one output layer with 1 neurons.	None	0.001	Relu for input layer, tanh for the LSTM hidden layers and linear for the output layer	Adam	MSE	MAE, MAPE, RMSE	Yes (patient = 10)
CNN	Sequential	model.add(Conv1D(filters=10, kernel_size = 3, padding='same', activation='relu', input_shape=(look_back, n_features))) model.add(MaxPooling1D(1)) model.add(Flatten()) model.add(Dense(1)) model.add(Activation('linear')) model.compile(loss='mean_squared_error', optimizer=keras.optimizers.Adam(learning_rate=LEARNING_RATE), metrics=['mae', 'mape'])	One input 1D Convolutional layer with 10 neurons, 3x3 kernel, one MaxPooling layer and one flatten layer. Three hidden layers consisting of 1D CNN layers with 30, 20, 10 neurons respectively. One output layer with One neuron.	None	0.001	Relu for input layer and linear for the output layer	Adam	MSE	MAE, MAPE, RMSE	Yes (patient = 10)
CNN LSTM FCNN	Sequential	model.add(Conv1D(filters=60, kernel_size = 3, padding='same', activation='relu', input_shape = (look_back, n_features))) model.add(MaxPooling1D(1)) model.add(LSTM(128, activation='tanh', return_sequences=True)) model.add(Dropout(DROP_OUT)) model.add(Dense(150, activation = 'relu')) model.add(Dense(10, activation = 'relu')) model.add(Flatten()) model.add(Dense(1)) model.add(Activation('linear')) model.compile(loss='mean_squared_error', optimizer=keras.optimizers.Adam(learning_rate=LEARNING_RATE), metrics=['mae', 'mape'])	One input 1D Convolutional layer with 10 neurons, 3x3 kernel, one MaxPooling layer and one flatten layer. Three hidden layers consisting of LSTM layer with 128 neurons and dropout, LSTM hidden layer with 32 nodes, one FCNN hidden layer with 10 neurons and One output layer with One neuron.	None	0.001	Rlu for input, tanh for hidden layers and linear for output	Adam	MSE	MAE, MAPE, RMSE	Yes (patient = 10)
CNN BLSTM FCNN	Sequential	model.add(Conv1D(filters=10, kernel_size = 3, padding='same', activation='relu', input_shape=(look_back, n_features))) model.add(MaxPooling1D(1)) model.add(Bidirectional(LSTM(50, activation='relu'))) model.add(Dense(50, activation = 'relu')) model.add(Dense(10, activation = 'relu')) model.add(Flatten()) model.add(Dense(1)) model.add(Activation('linear')) model.compile(loss='mean_squared_error', optimizer=keras.optimizers.Adam(learning_rate=LEARNING_RATE), metrics=['mae', 'mape'])	One input 1D Convolutional layer with 10 neurons, 3x3 kernel, one MaxPooling layer and one flatten layer. Three hidden layers consisting of Bidirectional LSTM layer with 150 neurons and dropout, FCNN with 150 nodes, one mode FCNN hidden layer with 10 neurons and One output layer with One neuron.	None	0.001	Rlu for input, tanh for hidden layers and linear for output	Adam	MSE	MAE, MAPE, RMSE	Yes (patient = 10)

The following is a brief description of DL techniques that were used to construct the evaluated models:

1) Fully Connected Neural Networks (FCNN)

In the current study, this model was optimized to make it a hard baseline target to benchmark other models against. Data inputs are the previous day's closing price, with number of values to consider determined by forecasting window parameters. The output is the close forecast price.

Various combinations of the number of layers, the number of nodes inside each layer, and key hyperparameters values were explored to determine the optimal network architecture. Python *Keras* library was used to accomplish this task. As shown in Table 3.1, the FCNN model was made of three hidden layers and the input and output layers. The hidden layers consisted of 50, 50, 25 neurons with a “relu” activation function, an Adam optimizer, and a loss function based on the "mean squared error". After trial and error and with the Keras GridSearch function's help to find the optimal hyperparameters' values, an acceptable performance for a baseline model was achieved. The optimal hyperparameters are displayed in table 3.1. Figure 3.3 shows a general architecture of an FCNN model.

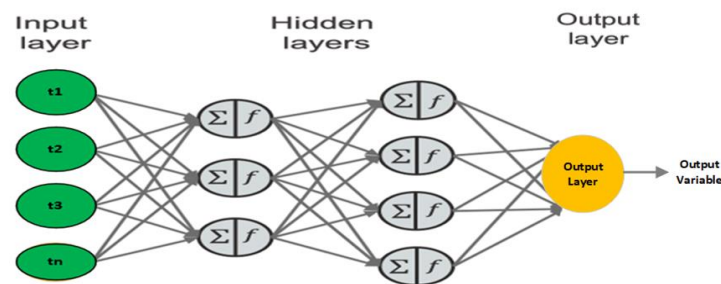


Figure 3.3 FCNN Model Architectures (source: Google Images)

2) Long Short-Term Memory Neural Network (LSTM)

Figure 3.4 shows a general architecture of an LSTM technique. As described in Table 3.1, This model consisted of three hidden LSTM layers and a densely connected output layer with one neuron. The model was configured with a "tanh" activation function, 0.2 dropout, 50 epochs Adam optimizer with 0.001 learning rate, and a loss function based on the "mean squared error". This model was manually optimized by trying different architectures, hyperparameters values, and a different number of epochs.

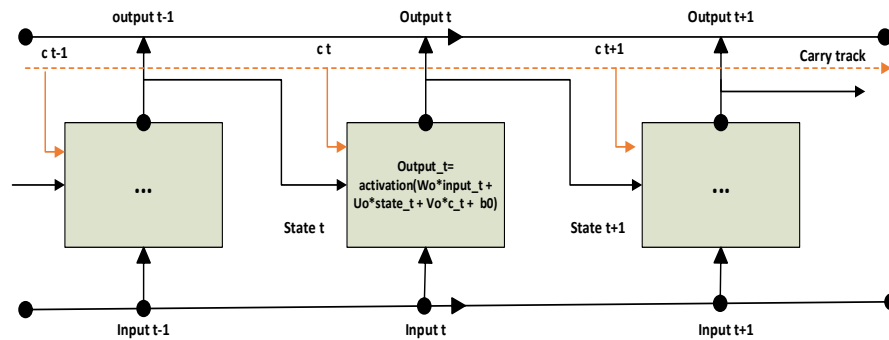


Figure 3.4 LSTM Architectures (source: Google Images)

3) Bidirectional LSTM

Figure 3.5 shows a general architecture of the BLSTM technique. This model consisted of three hidden BLSTM layers with 128, 64, 32 units, respectively, and a densely connected output layer with one neuron. This model was configured with a "tanh" activation function, 0.2 dropout, 50 epochs Adam optimizer with 0.001 learning rate, and a loss function based on the "mean squared error".

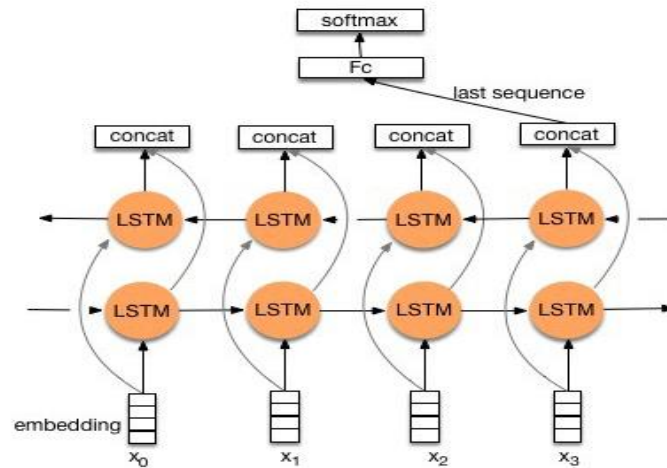


Figure 3.5 BLSTM General Architectures (source: Google Images)

4) Convolutional Neural Network (CNN)

Figure 3.6 shows a general architecture of the CNN model architecture. In this research 1D CNN was used to offer a fast alternative to LSTM for trivial tasks such as time series forecasting. In Keras, 1D CNN is configured via the Conv1D layer, which has an interface like Conv2D. This model consisted of three hidden 1D CNN layers with 30, 20, 10 filters, respectively, and a densely connected output layer with one unit. Furthermore, the "relu" activation function, 0.2 dropouts, 50 epochs Adam optimizer with 0.001 learning rate, and a loss function based on the "mean squared error" were used to configure this model.

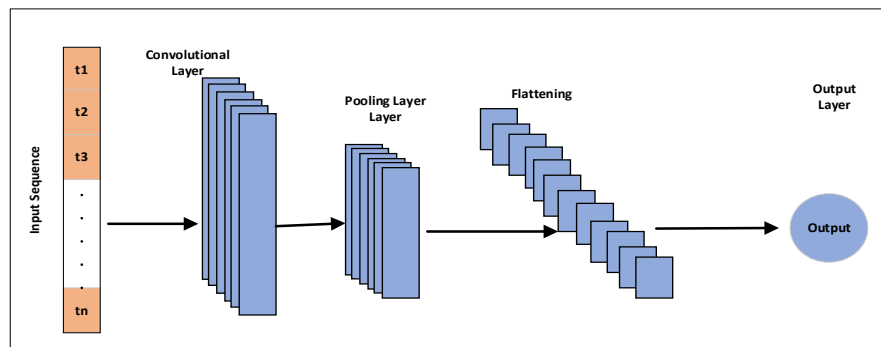


Figure 3.6 Convolutional 1D model Architectures

5) Combined Model 1

The general architecture of the first combined model is shown in Figure 3.7. In the model, 1D CNN was used as a preprocessing step for downsampling and extracting higher-level features, which can then be passed on to the LSTM. LSTM was added to the model for the time sequence prediction. LSTM layer is fed inputs after 1D CNN convolution with the actual data features, and subsequently the pooling process is performed. As a result, complex calculations were avoided. The objective of this design was to combine the speed and lightness of CNNs with the order sensitivity of LSTM. Using 1D CNN as a preprocessing step before an LSTM is a techniques that is not seen often in research papers and practical applications, possibly because it is not well known.

This model was constructed by combining one input 1D CNN layer with ten neurons, a 3x3 kernel, one MaxPooling layer, and one flattened layer. This layer was followed by three hidden LSTM layers with 128, 64, and 32 neurons, respectively, with 0.2 dropout rates. Two FCNN hidden layers with 50, 25 neurons, respectively, with 0.2 dropout rates, were placed just before the output layer.

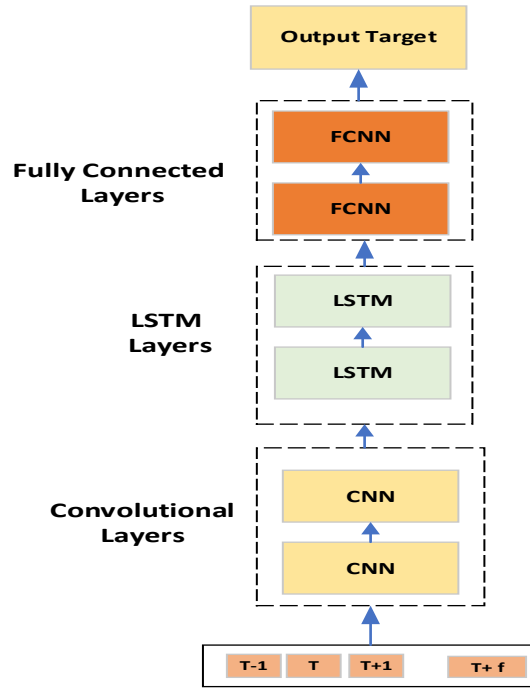


Figure 3.7 Hybrid Model 1 Architectures

6) Combined Model 2

Figure 3.8 shows the general architecture of the second combined model. In this model, LSTM layers were replaced with BLSTM layers. The output of the BLSTM layer is fed into a fully connected layer, which is flattened and fed into an output layer with one neuron. This model was constructed using one input 1D CNN layer with ten neurons, 3x3 kernel, one MaxPooling layer, and one flattened layer. Three hidden layers consisted of a Bidirectional LSTM layer with 150 neurons and dropout, FCNN with 150 nodes, one more FCNN hidden layer with ten neurons, and One output layer with one neuron.

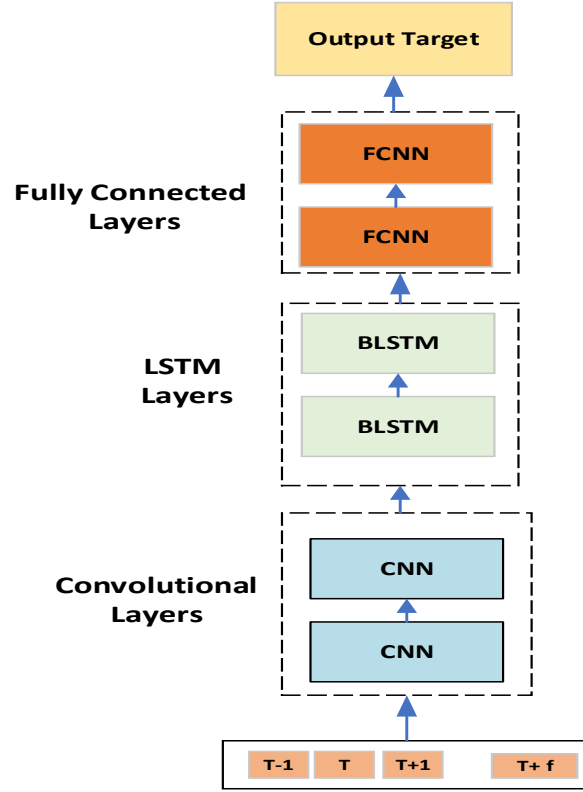


Figure 3.8 Hybrid Model 2 Architectures

3.2.2 Evaluation Metrics

To compare the performance of the implemented models, more than one metric has been used, each with its own advantages and disadvantages since there is no universal metric that can lead to an overall conclusion about the performance of the models. The evaluation approach used three metrics, which are widely used for evaluating forecasting techniques. These are: MAE, RMSE, and MAPE (Tsihrintzis et al, 2018). The performance metrics can be defined via the following formulas using y as the actual value and $\hat{y}$ as the predicted value:

- 1) MAE is the arithmetic average of absolute errors, and it is computed by the following formula:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}| \quad (1)$$

It is impossible to use MAE to compare series of different scales, as it is scale-dependent. It is, however, commonly used for making comparisons between different models when the data is homogeneous. Its robustness to outliers makes it an attractive measure.

- 2) In forecasting, the RMSE means the squared root of the average of the squares of the error. It can be computed as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y})^2} \quad (2)$$

Since the RMSE is also scale-dependent, it cannot be used for comparing multiple datasets. RMSE is, however, appropriate for comparing more than one model on a single dataset. As the squared difference in its formula can significantly increase its value due to the presence of outliers, the RMSE is sensitive to outliers. As RMSE is always a positive value, lower RMSE values indicate better forecast performance, while zero suggests a perfect forecast.

- 3) MAPE is a measure of the average of absolute percentage errors computed using a formula:

$$\text{MAPE} = \left\{ \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}}{y_i} \right| \right\} * 100 \quad (3)$$

It is one of the most commonly used methods for comparing forecasting methods, as it is easily interpreted, alongside being scale-independent. When the prediction value is much lower or higher than the actual value, the MAPE can score values over 100, which is referred to as a percentage error. Furthermore, MAPE is not defined when the actual

value equals zero (Chowdhury et al., 2020). Such values did not occur in this study. The use of MAPE is thus recommended. To determine the accuracy of a forecast, MAPE is used as follows:

$$\text{Forecasting Accuracy} = 100 - \text{MAPE} \quad (4)$$

This study also utilizes the learning curves evaluation tool to understand and explain how the model training process is performing. Learning curves are commonly used in machine learning for algorithms that improve incrementally over time, such as deep learning neural networks. In this context, the metric used to assess learning could be maximized, which means that better scores (more numbers) indicate greater learning. At each step of the training algorithm, the model's current state can be evaluated. Evaluation can be conducted on the training dataset to judge how well the model is performing. Evaluation can also be conducted on a validation dataset that is separate from the training dataset. The validation dataset can provide insight into the generalization capacity of the model.

An indication of how well a model is learning based on the training dataset is called the train learning curve. A validation learning curve gives an indication of how well the model is generalizing based on a hold-out validation dataset. During the training of a machine learning model, dual learning curves are commonly created for both the training and validation datasets. Three common dynamics can be observed in learning curves, and they are:

Underfitting: Underfitting occurs when a model cannot learn from the training data. The model is underfitted if it is unable to obtain a sufficiently low error value on the training set. The learning curve of the

training loss alone can be used to identify an underfit model. The loss curve may be flat or noisy, indicating that the model was unable to learn the training set.

Overfitting: Overfitting refers to a model that has learned the training dataset too well, including statistical noise and random fluctuations. Overfitting results in an increase in generalization gap, since the more specialized a model becomes to training data, the harder it is for it to generalize to new data. The model's performance on the validation dataset can provide information about how much generalization gap has increased.

Good Fit: This is the goal of the learning algorithm, to get a good fit between overfit and underfit models. If the training and validation losses decrease to a point of stability with a minimal gap between the two final loss values, then the fit is good. Losses for the model are always lower on training data than on validation data. Therefore, there will be some gaps between the train loss and validation loss learning curve. This gap is referred to as the “generalization gap.”

Chapter Four:

Experimentations and Results

The experimentations and results chapter is structured first to outline the conducted experiments described in section 4.1. Section 4.2 provides detailed descriptive statistics of the selected two cryptocurrencies, Bitcoin and Ether. Next, it presents the results for the base model, FCNN, performance. Section 4.3 describes the performance of the FCNN model. Sections 4.4 to 4.8 describe the results of the other five models' performance compared to the base model. Section 4.9 presents the overall results of comparing all models with the base model to identify the best model that yielded the best rest for each cryptocurrency.

4.1 Experiment Outline

Figure 4.1 depicts the pseudo-code of the program developed using Python for each model of the previously described DL models. The experiment was repeated ten times for each model using the lags, look backs, list, and applied on both cryptocurrencies Bitcoin and Ether. Before supervised learning can be performed, time series data must be transformed. The dataset was restructured to be appropriate for a supervised learning problem by using the value at the previous time lag as the lookback windows to predict the value at the next time step using the forecast window. In the sliding window method, past time steps are used to predict the future step. Likewise, the size of the lag is calculated by the number of previous time steps. Any time series dataset can be transformed into a supervised learning problem using this sliding window (Brownlee, 2017; Lazzeri, 2020).

In this study, the lag windows size was made equal to the forecast windows size and used the one-step prediction method. An observation at the current time step is the result of using observations at previous time steps, called lag observations. Once the time series datasets were prepared for the DL models, they were split into 80% training and 20% testing sets. DL models were fitted and tested using the train and test datasets. The loss function was monitored using the mean squared error. Early stopping was used to regulate once the best model is found; measurements metrics are recorded and saved in a separate file for each experiment.

```

Repeat the experiment for 10 times:
Experiment:
  For each ticker in ticker list [Bitcoin, Ether]:
    Fetch the cryptocurrency dataset from Yahoo Finance
    For each lookback in look_back_list [7, 14, 21, 28, 35, 42, 49, 56]:
      Generate a time series dataset for the selected ticker and lookback
      Split the datasets into 80% training and 20% test
      Initialize the model name of 6 models, separate code file per model name
      Train the model using the training dataset
      Fit and validate each model of the six model on the test dataset
        Implement early stopping and save the best model
        Load the best model
        Evaluate model performance
        Save training and test results
        Implementing the model for forecasting the future values:
      Download the latest real ticker close price for the required forecast period
      Load and use the best model for forecasting
      Compare True close price to forecasted close price by calculating MAE, MAPE
      and RMSE metrics
      Save the forecasting results
    Repeat for the next lookback Value in the look_back_list
  Repeat for the next ticker in the ticker list
Repeat the experiment

```

Figure 4.1 Experiment Pseudo Algorithm

Finally, the best model found for each forecasting horizon was loaded and used iteratively to perform multi-step forecasting equal to the forecasting window size. Furthermore, the experiment was repeated for

each forecasting window size used in this experiment. After conducting the experiment 10 times, all results were saved in a file and used for comparison.

4.2 Descriptive Statistics

Bitcoin

Bitcoin dataset consisted of 1,614 observations and one attribute as shown in Table 4.1. This attribute is:

- Close: the last price at which the Bitcoin traded during the regular trading day.

Table 4.1 shows a statistical summary of the Bitcoin close price, and Figure 4.2 shows a bar chart presentation of this attribute. Figure 4.3 depicts the close price over time for the selected period. In Table 4.1, the Close Price column is highlighted because it is the only feature that has been used in the study as a univariate time series forecasting problem. Therefore, only this column was retained, and all other columns were dropped.

Table 4.1 Bitcoin Close Price Descriptive Statistics

	Close
count	1,614
mean	15,313.206
std	15,689.802
min	1,929.820
25%	6,454.435
50%	8,896.524
75%	13,628.286
max	65,992.836

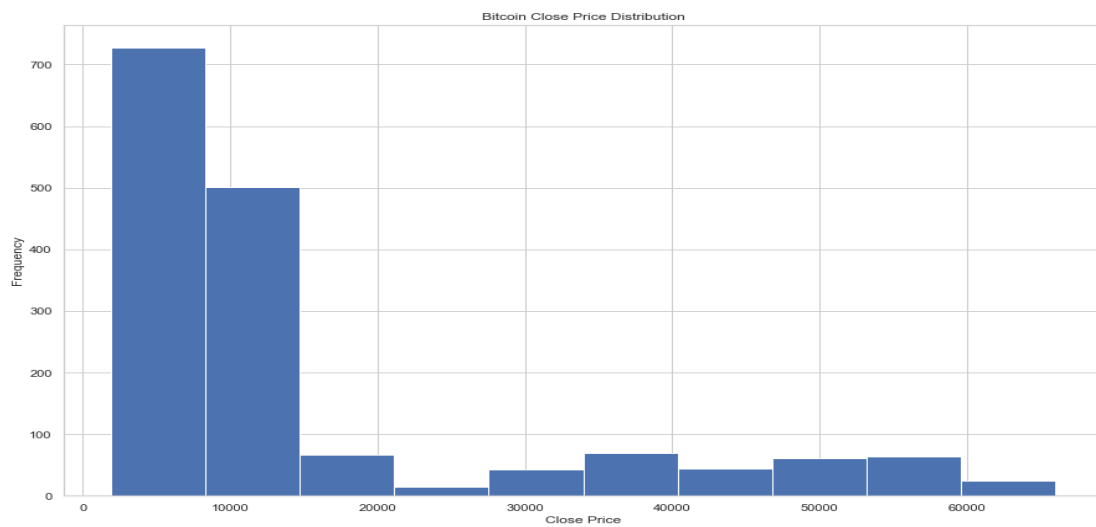


Figure 4.2 Bitcoin Close Price Frequency Distribution



Figure 4.3 Bitcoin Close Price Time Series

Ether

Ether dataset consisted of 1,452 data points and one attribute described in Table 4.2. This attribute is:

- Close: the last price at which the Ether traded during the regular trading day.

Table 4.2 shows a statistical summary of the Ether close price, and Figure 4.4 shows a bar chart presentation of this attribute. Figure 4.5 depicts the close price over time for the selected period. Like the Bitcoin dataset, only the close price attribute was retained.

Table 4.2 Ether Close Price Descriptive Statistics

	Close
count	1,452
mean	781.672
std	976.287
min	84.308
25%	186.545
50%	332.876
75%	816.328
max	4414.747

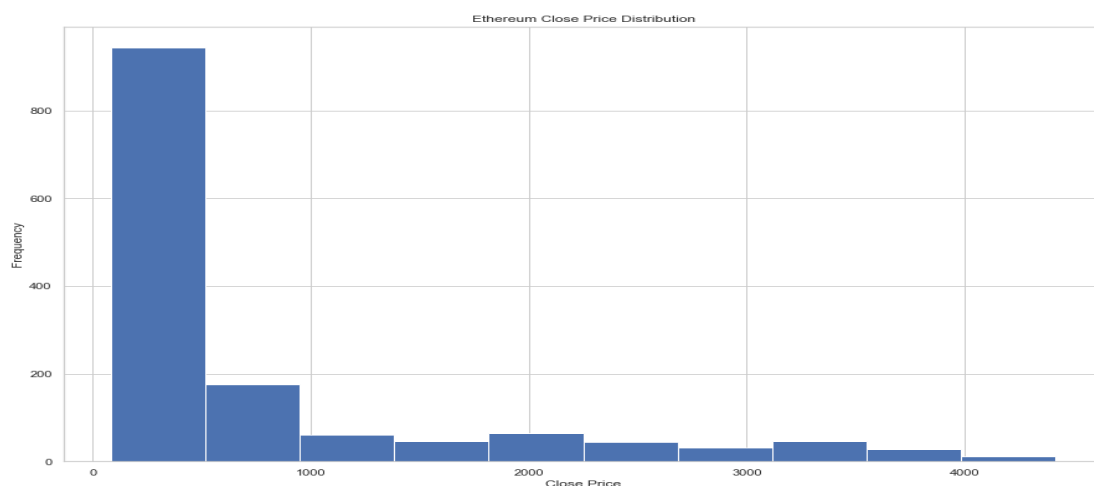


Figure 4.4 Ether Close Price Frequency Distribution



Figure 4.5 Ether Close Price Time Series

4.3 Base Model FCNN Performance

FCNN was selected to be the baseline model because fully connected feedforward networks can be easily constructed and optimized for the best performance. Therefore, it was imperative to optimize this model performance until it became a hard target to outperform.

FCNN Model Fitting and Testing Results

Figures 4.6(a) and (b) show the results of training and fitting the FCNN models on both Bitcoin and Ether, respectively, using a forecasting window of 7 days. This model is learning from the training datasets, as shown by the blue line in the graph. The validation, or test, the orange line shows the model generalization capacity. This gap is referred to as the "generalization gap" (Brownlee, 2018). Figures 4.7(a) and (b) show the best model deployment performance. It compares forecasted values with the actual values for Bitcoin and Ether, respectively. The last 350 data points of the close price time series were shown for graph clarity and straightforward interpretation. A complete list of graphs for this model's performance on other sliding windows can be found in Appendix A.

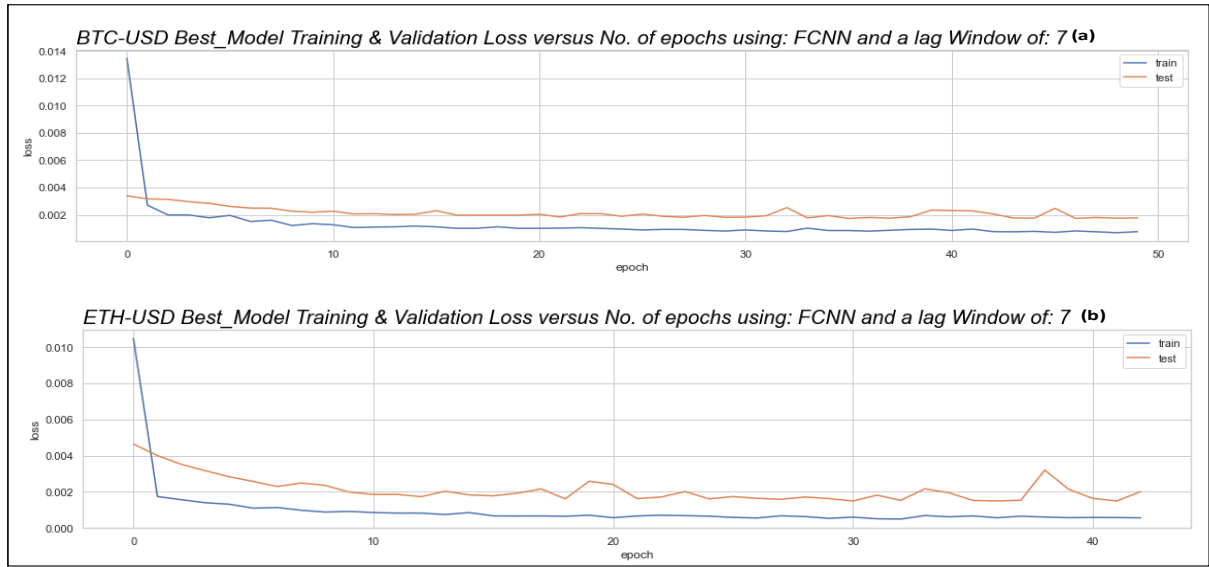


Figure 4.6 FCNN Model Fitting and Validation Results

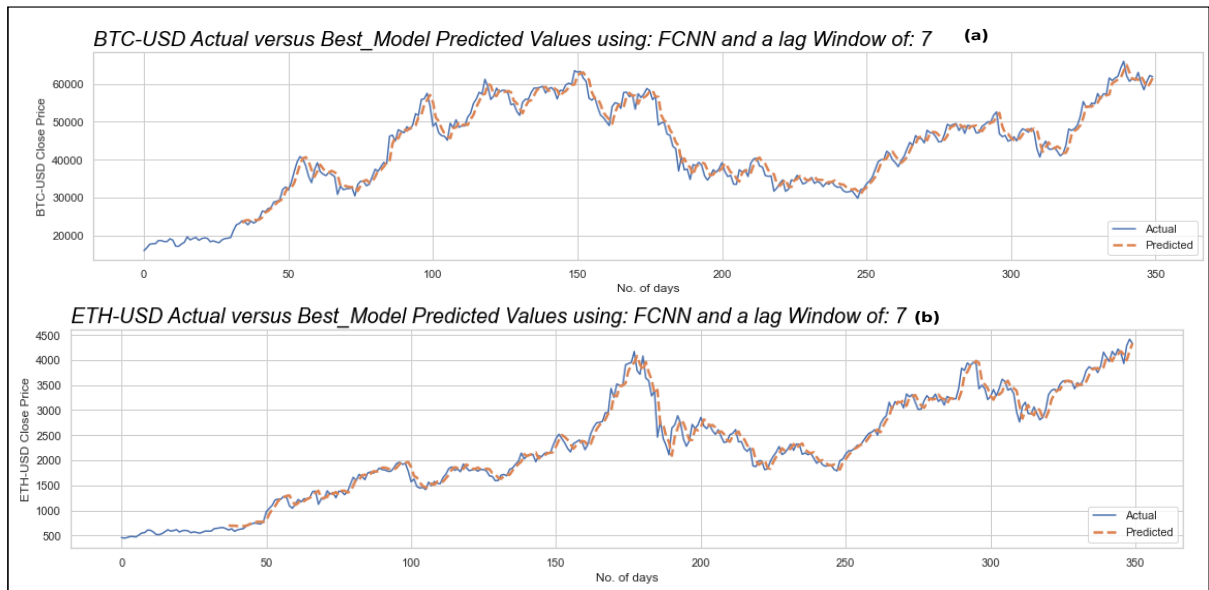


Figure 4.7 FCNN Model Prediction Performance

Table 4.3 displays the model forecasting performance metrics for each cryptocurrency and forecasting windows. Figure 4.8 (a) and (b) show the forecasting results for seven days forecasting windows against the actual close price, which are downloaded as soon they are published on the Yahoo Finance site for Bitcoin and Ether, respectively. The actual close

prices are being used to compare the model's accuracy when deployed in a real-life environment.

Table 4.3 FCNN Model Forecasting Performance

Ticker	Forecasting Windows	Forecasting mae	Forecasting mape	Forecasting rmse	Forecasting accuracy %
BTC-USD	7	1034.86	1.99	1230.99	98.01
	14	2175.30	4.06	2644.16	95.94
	21	2895.15	5.91	3648.56	94.09
	28	3812.29	8.58	4517.06	91.42
	35	5762.85	12.58	7437.53	87.42
	42	6098.32	14.98	7569.70	85.02
	49	5306.14	15.55	7051.97	84.45
	56	4923.56	15.63	6616.87	84.37
ETH-USD	7	325.91	7.22	342.04	92.78
	14	557.23	12.07	602.68	87.93
	21	647.84	14.26	689.44	85.74
	28	659.16	14.65	705.03	85.35
	35	792.36	17.78	879.17	82.22
	42	859.62	19.47	951.30	80.53
	49	820.99	18.88	911.49	81.12
	56	749.83	17.53	853.08	82.47

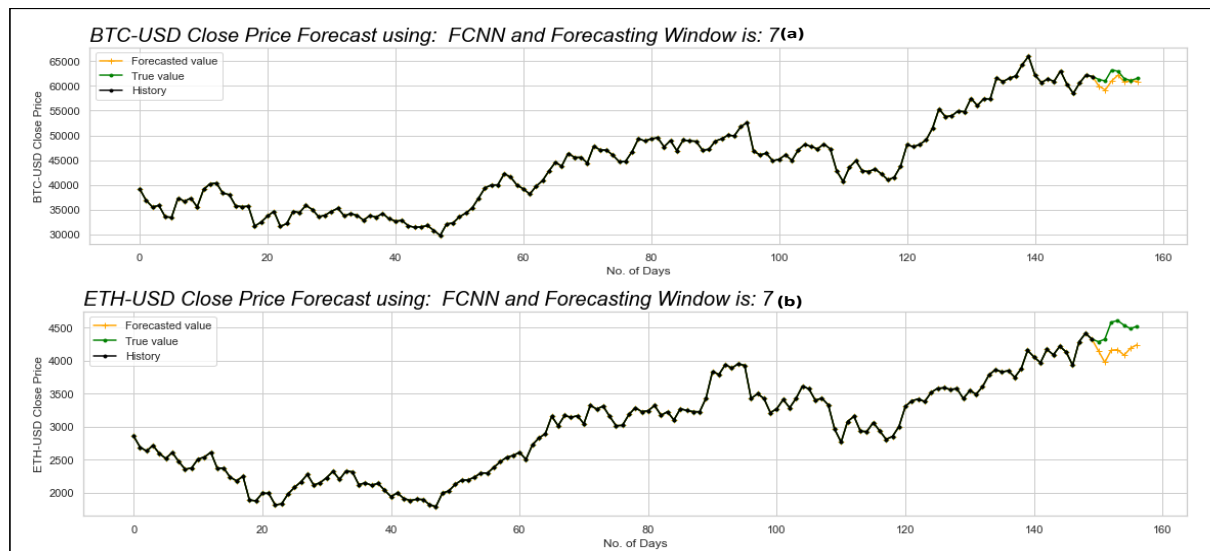


Figure 4.8 FCNN Model Forecasting Performance

FCNN Model Forecasting Performance

Figure 4.9 depicts the FCNN model forecasting performance metrics MAPE, RMSE, and Accuracy for both cryptocurrencies against the forecasting window size. As mentioned above, the model performance for each forecasting window is the average forecasting performance of the ten times that the experiment was conducted. This was applied to all other models' results.

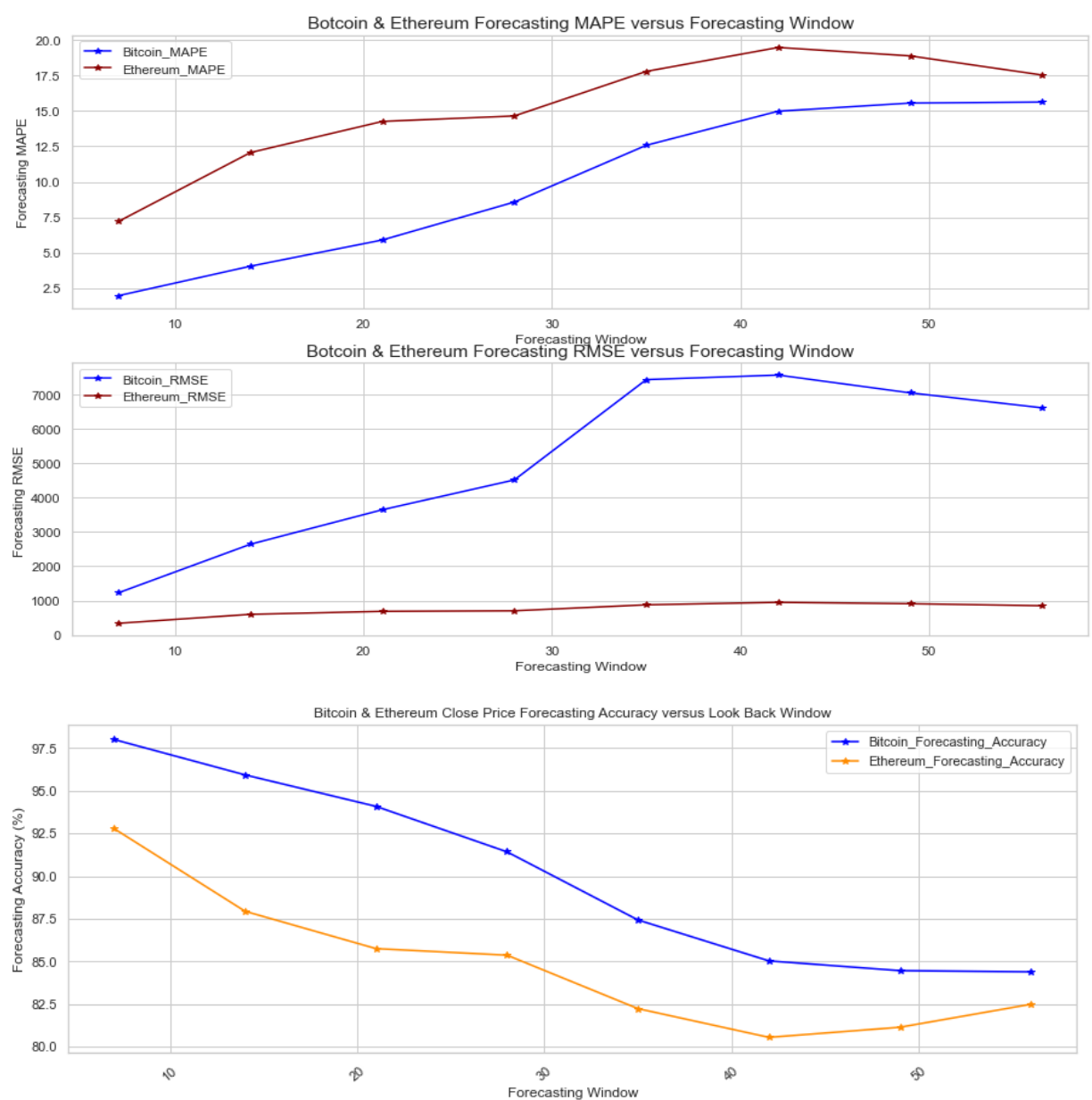


Figure 4.9 FCNN Model Forecasting Performance Metrics

4.4 LSTM Model Performance

LSTM Model Fitting and Testing Results

Figure 4.10 (a) and (b) depict the LSTM model training and validation results on Bitcoin and Ether close price. The results shown in these figures are for a forecasting window of 7 days. Appendix A shows the rest of the forecasting windows used.

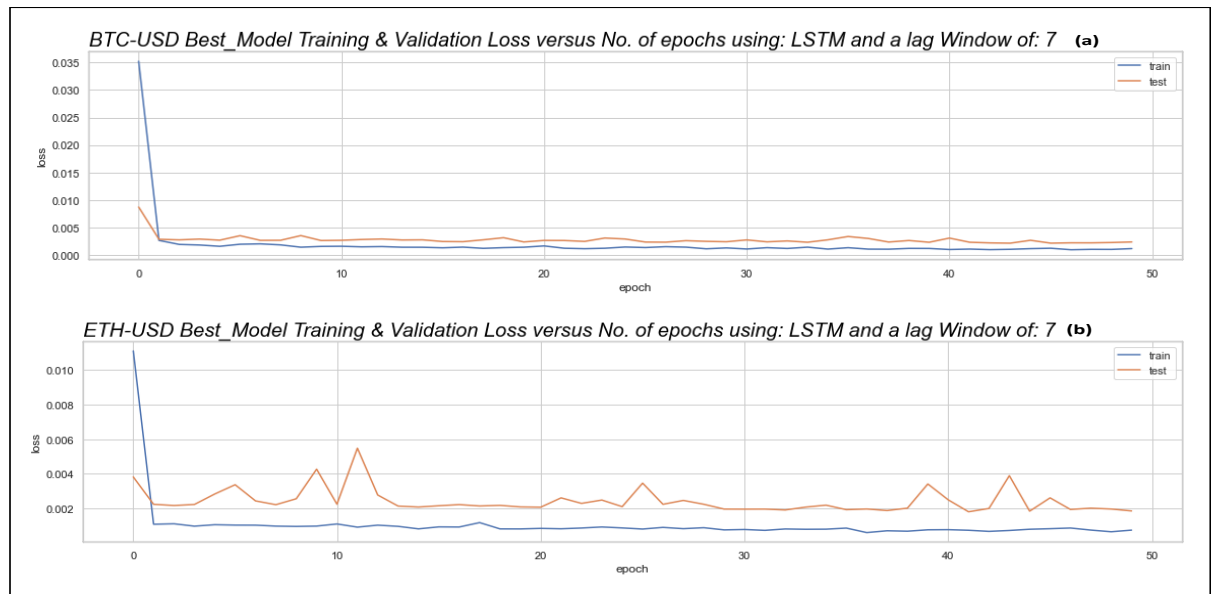


Figure 4.10 LSTM Model Fitting and Validation Results

Figures 4.11(a) and (b) show the best LSTM model prediction performance versus the actual values for Bitcoin and Ether, respectively. A complete list of graphs for this model performance on other forecasting windows are provided in Appendix A.

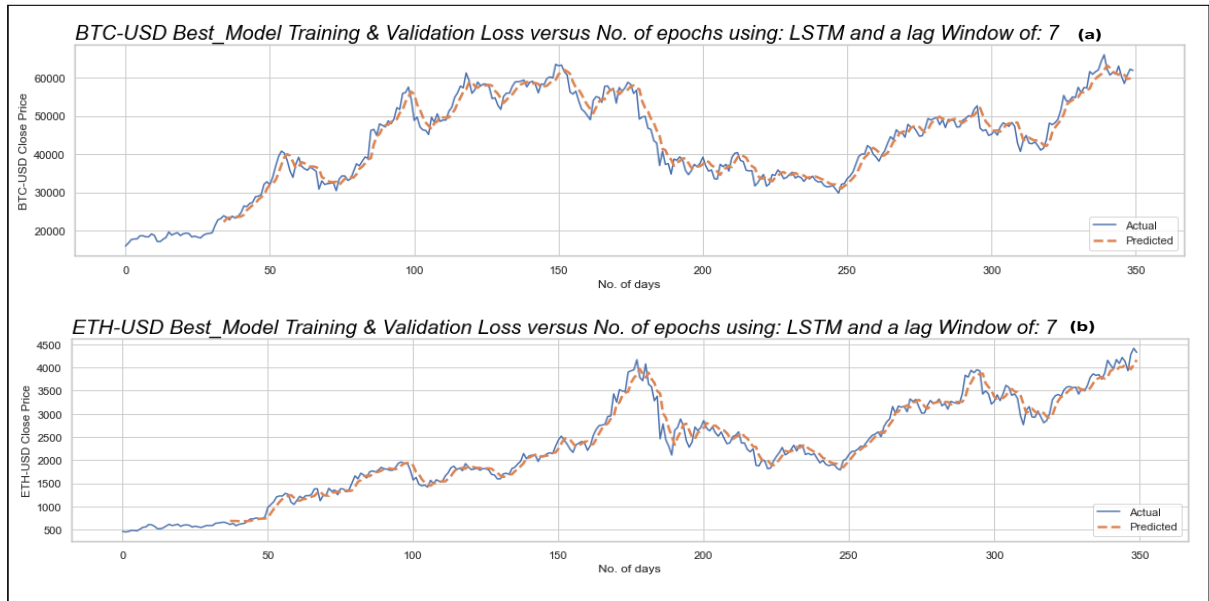


Figure 4.11 LSTM Model Prediction Performance

Table 4.4 displays the LSTM model forecasting performance metrics for each cryptocurrency and forecasting window compared to the base FCNN model.

Table 4.4 LSTM Model Forecasting Performance

Ticker	Forecasting Window	Forecasting MAE		Forecasting MAPE		Forecasting RMSE		Forecasting Accuracy %	
		FCNN	LSTM	FCNN	LSTM	FCNN	LSTM	FCNN	LSTM
BTC-USD	7	1034.86	1197.11	1.99	2.03	1230.99	1316.08	98.01	97.97
	14	2175.30	2270.74	4.06	3.86	2644.16	2789.11	95.94	96.14
	21	2895.15	2822.10	5.91	5.56	3648.56	3697.49	94.09	94.44
	28	3812.29	3837.81	8.58	8.19	4517.06	4441.06	91.42	91.81
	35	5762.85	5831.41	12.58	12.04	7437.53	7245.29	87.42	87.96
	42	6098.32	6339.55	14.98	14.81	7569.70	7784.86	85.02	85.19
	49	5306.14	5561.13	15.55	15.35	7051.97	7186.71	84.45	84.65
	56	4923.56	5028.58	15.63	15.36	6616.87	6631.13	84.37	84.64
ETH-USD	7	325.91	434.11	7.22	9.63	342.04	447.00	92.78	90.37
	14	557.23	685.97	12.07	14.87	602.68	716.89	87.93	85.13
	21	647.84	695.24	14.26	15.28	689.44	733.71	85.74	84.72
	28	659.16	722.17	14.65	16.10	705.03	759.12	85.35	83.90
	35	792.36	853.48	17.78	19.11	879.17	918.85	82.22	80.89
	42	859.62	887.60	19.47	20.06	951.30	960.00	80.53	79.94
	49	820.99	847.37	18.88	19.44	911.49	928.58	81.12	80.56
	56	749.83	806.90	17.53	18.63	853.08	890.78	82.47	81.37

Figure 4.12 shows the forecasting results for seven days forecasting window against the actual close price of both currencies. The actual close prices are being used to compare the model's accuracy in forecasting future prices.

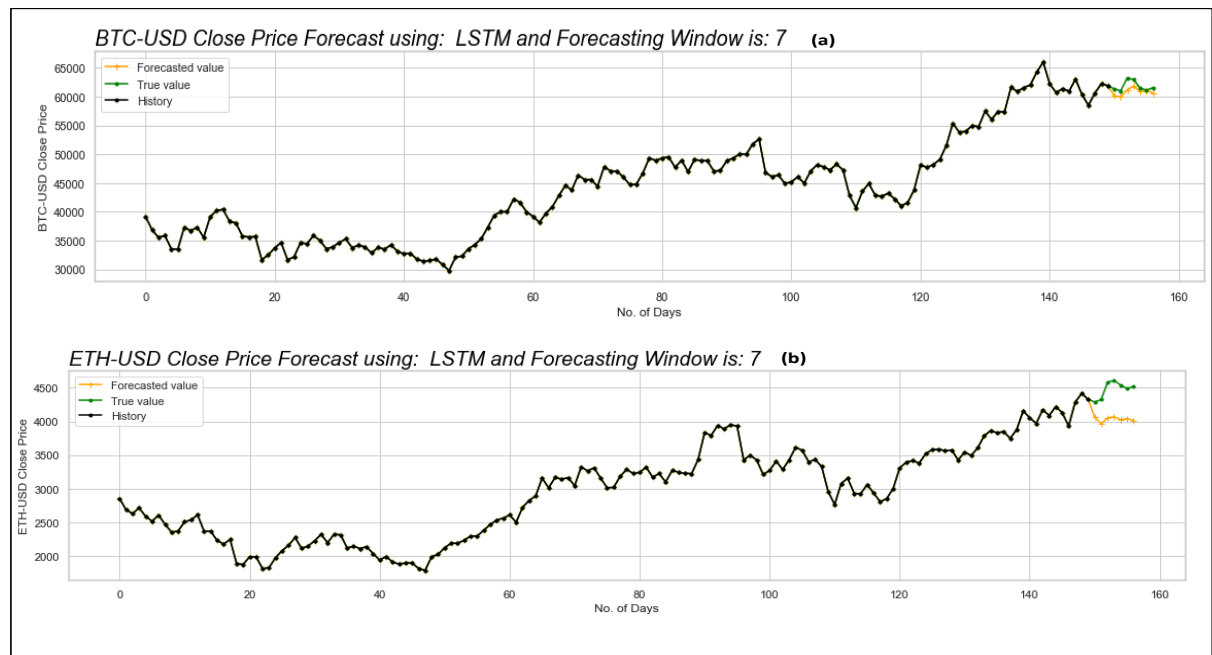


Figure 4.12 LSTM Model Forecasting Performance

LSTM Forecasting Performance

Figure 4.13 shows the LSTM model forecasting performance compared to the FCNN base model. As mentioned above, the model performance for each forecasting window is the average forecasting performance of the ten times that the experiment was conducted. This applies to all other models' results. For Bitcoin, the average scores across all forecasting windows for MAPE, RMSE, and Accuracy are 9.65, 5136.47, 90.35%, respectively. For Ether, LSTM scored 17.66, 794.37, 83.36% for MAPE, RMSE, and Accuracy, respectively.

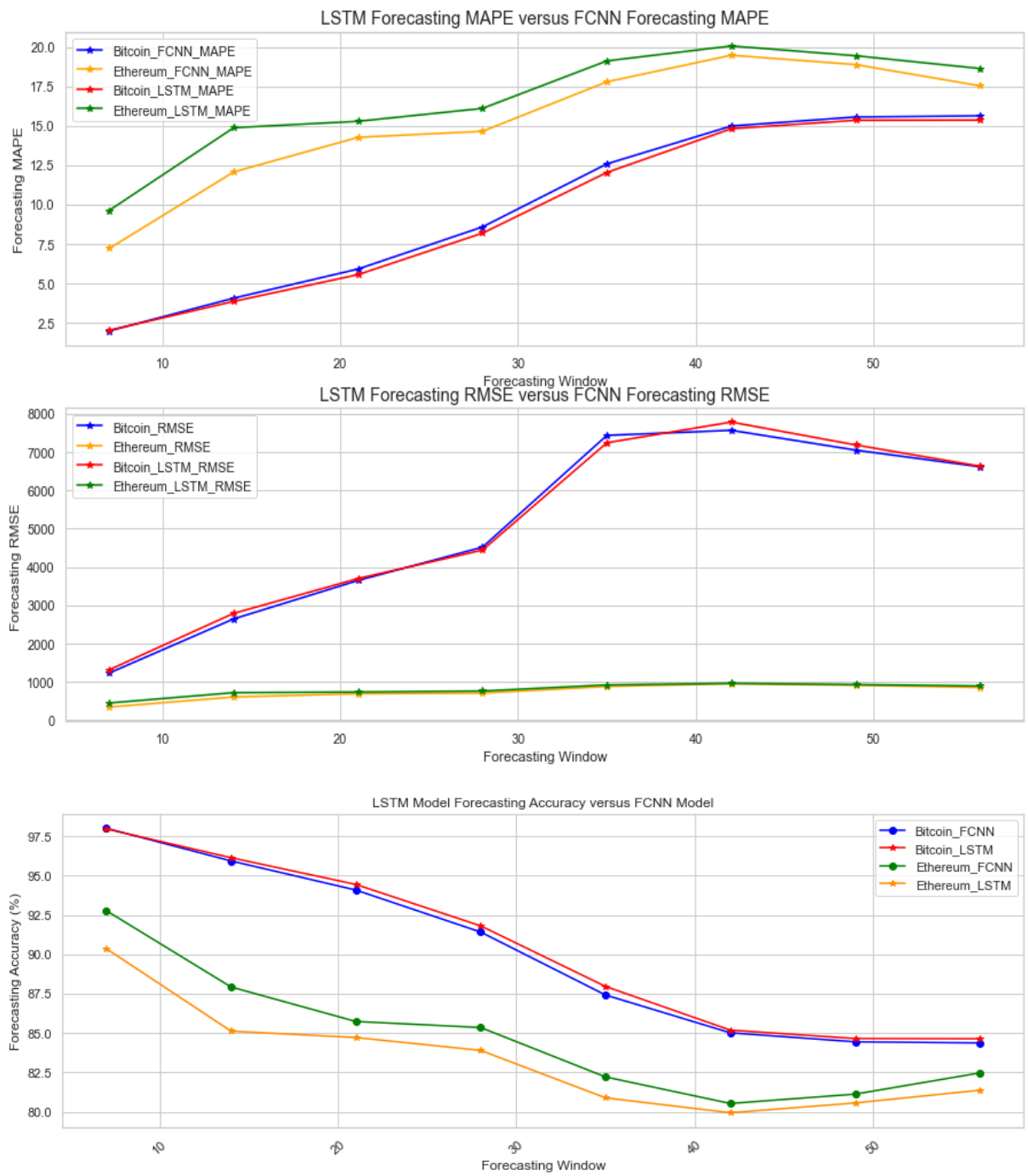


Figure 4.13 LSTM Model Forecasting Metrics Compared to FCNN

4.5 BLSTM Model Performance

Figure 4.14 (a) and (b) depict the BLSTM model training and validation results on Bitcoin and Ether close price dataset. The results shown in these figures are for a forecasting window of 7 days. Appendix A provides more information on the rest of the forecasting windows used.

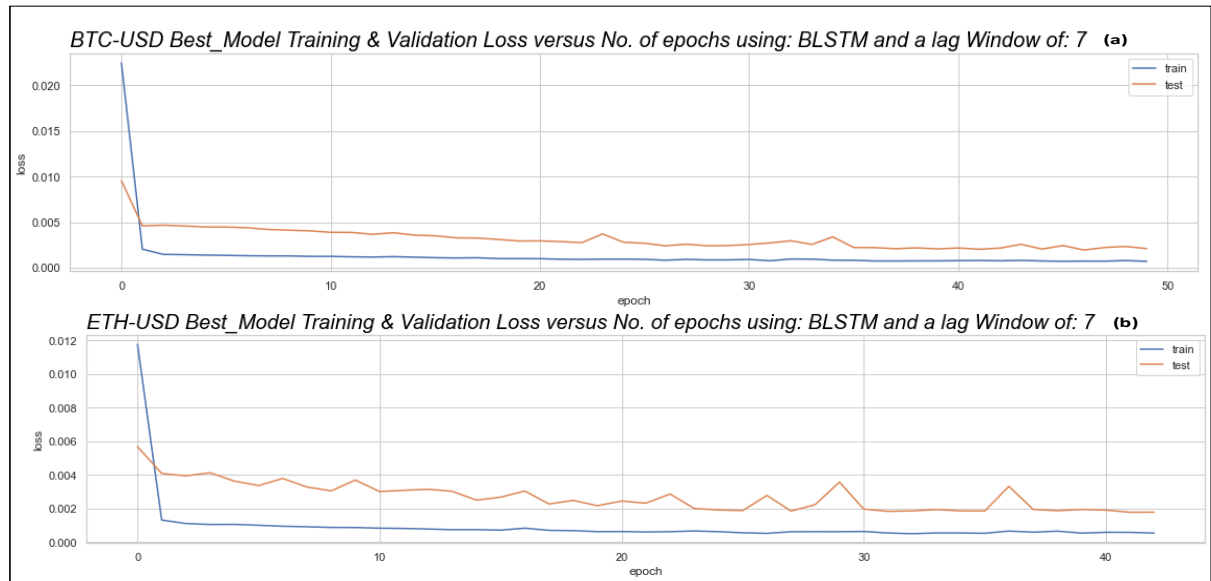


Figure 4.14 BLSTM Model Fitting and Validation Performance

Figures 4.15(a) and (b) show the best BLSTM model prediction performance versus the actual values for Bitcoin and Ether, respectively. A complete list of graphs for this model's performance on other forecasting windows are provided in Appendix A.

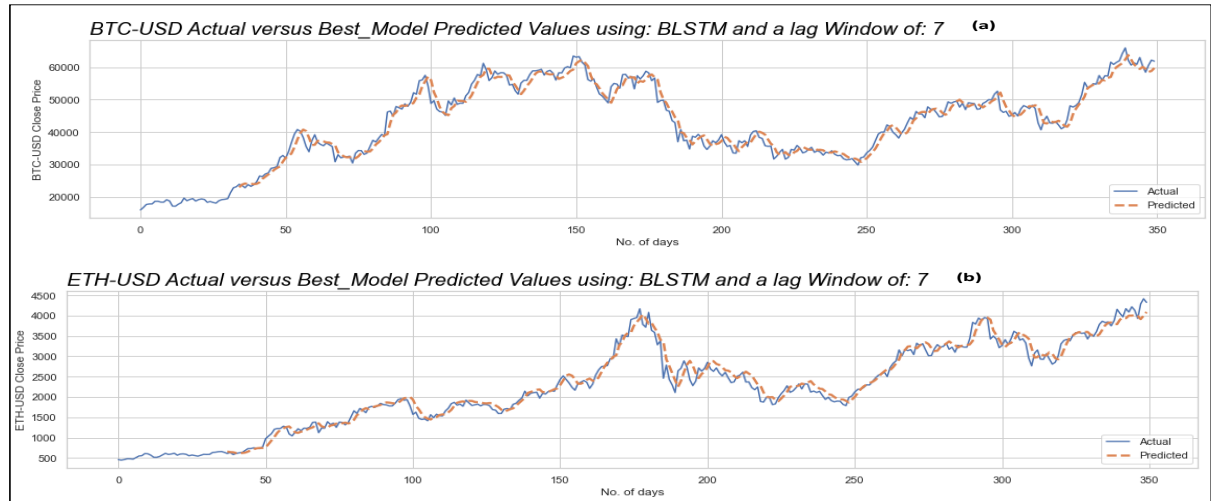


Figure 4.15 BLSTM Model Prediction Performance

Table 4.5 displays the BLSTM model forecasting performance metrics for each cryptocurrency and forecasting window compared to the base FCNN model.

Table 4.5 BLSTM Model Forecasting Performance

Ticker	Forecasting Window	Forecasting MAE		Forecasting MAPE		Forecasting RMSE		Forecasting Accuracy %	
		FCNN	BLSTM	FCNN	BLSTM	FCNN	BLSTM	FCNN	BLSTM
BTC-USD	7	1034.86	1744.03	1.99	2.88	1230.99	1800.02	98.01	97.12
	14	2175.30	2333.58	4.06	4.29	2644.16	2739.98	95.94	95.71
	21	2895.15	2633.73	5.91	5.65	3648.56	3287.77	94.09	94.35
	28	3812.29	3587.49	8.58	8.28	4517.06	4316.28	91.42	91.72
	35	5762.85	5683.72	12.58	12.08	7437.53	7084.23	87.42	87.92
	42	6098.32	5710.70	14.98	14.63	7569.70	7199.81	85.02	85.37
	49	5306.14	5238.35	15.55	15.24	7051.97	6813.63	84.45	84.76
	56	4923.56	4583.28	15.63	15.22	6616.87	6344.06	84.37	84.78
ETH-USD	7	325.91	510.39	7.22	11.34	342.04	520.18	92.78	88.66
	14	557.23	611.75	12.07	13.25	602.68	653.11	87.93	86.75
	21	647.84	661.50	14.26	14.55	689.44	695.98	85.74	85.45
	28	659.16	702.04	14.65	15.65	705.03	742.57	85.35	84.35
	35	792.36	818.64	17.78	18.35	879.17	899.68	82.22	81.65
	42	859.62	868.03	19.47	19.63	951.30	944.84	80.53	80.37
	49	820.99	832.04	18.88	19.07	911.49	911.74	81.12	80.93
	56	749.83	795.54	17.53	18.41	853.08	885.18	82.47	81.59

Figure 4.16 shows the forecasting results for seven days forecasting window against the actual close price of both currencies. The actual close prices are being used to compare the model's accuracy in forecasting future prices.

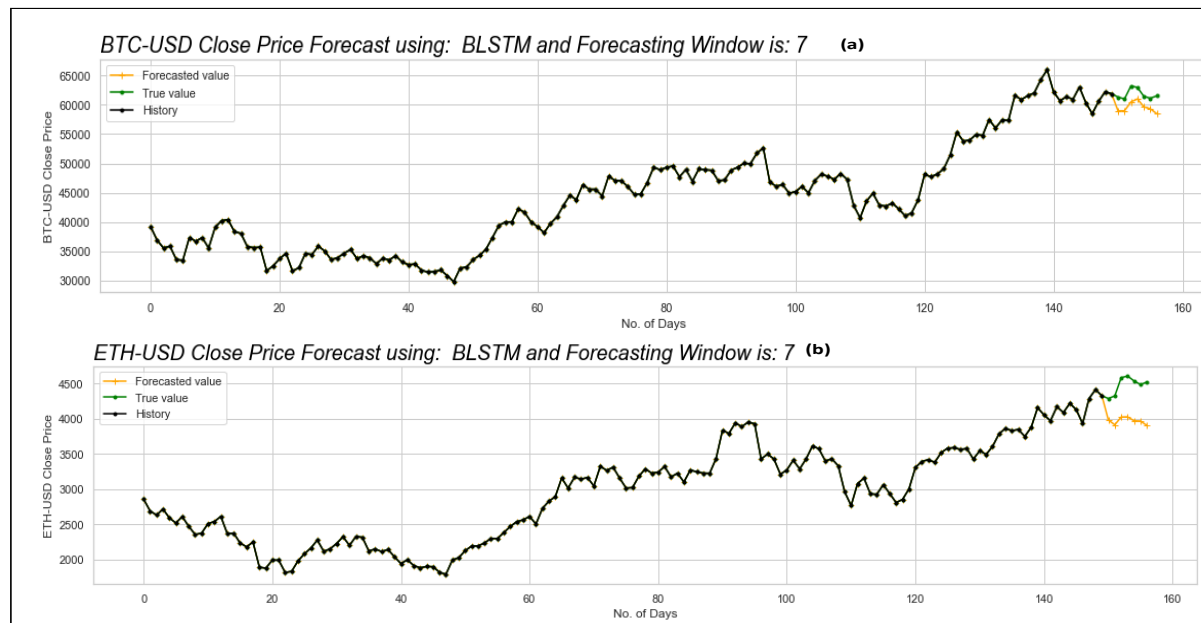


Figure 4.16 BLSTM Model Forecasting Performance

BLSTM Model Forecasting Performance

Figure 4.17 shows that for Bitcoin, the BLSTM model performed slightly better than the base model, FCNN, with a MAPE of 9.87, RMSE of 4948.22, and Accuracy of 90.22. However, these metrics declined as the forecasting windows increased. For Ether, BLSTM scored a MAPE of 17.03, RMSE of 781.66, and Accuracy of 83.72%. Although the model forecasting accuracy for Ether is lower than the base model score of 84.77%, it is insignificant.

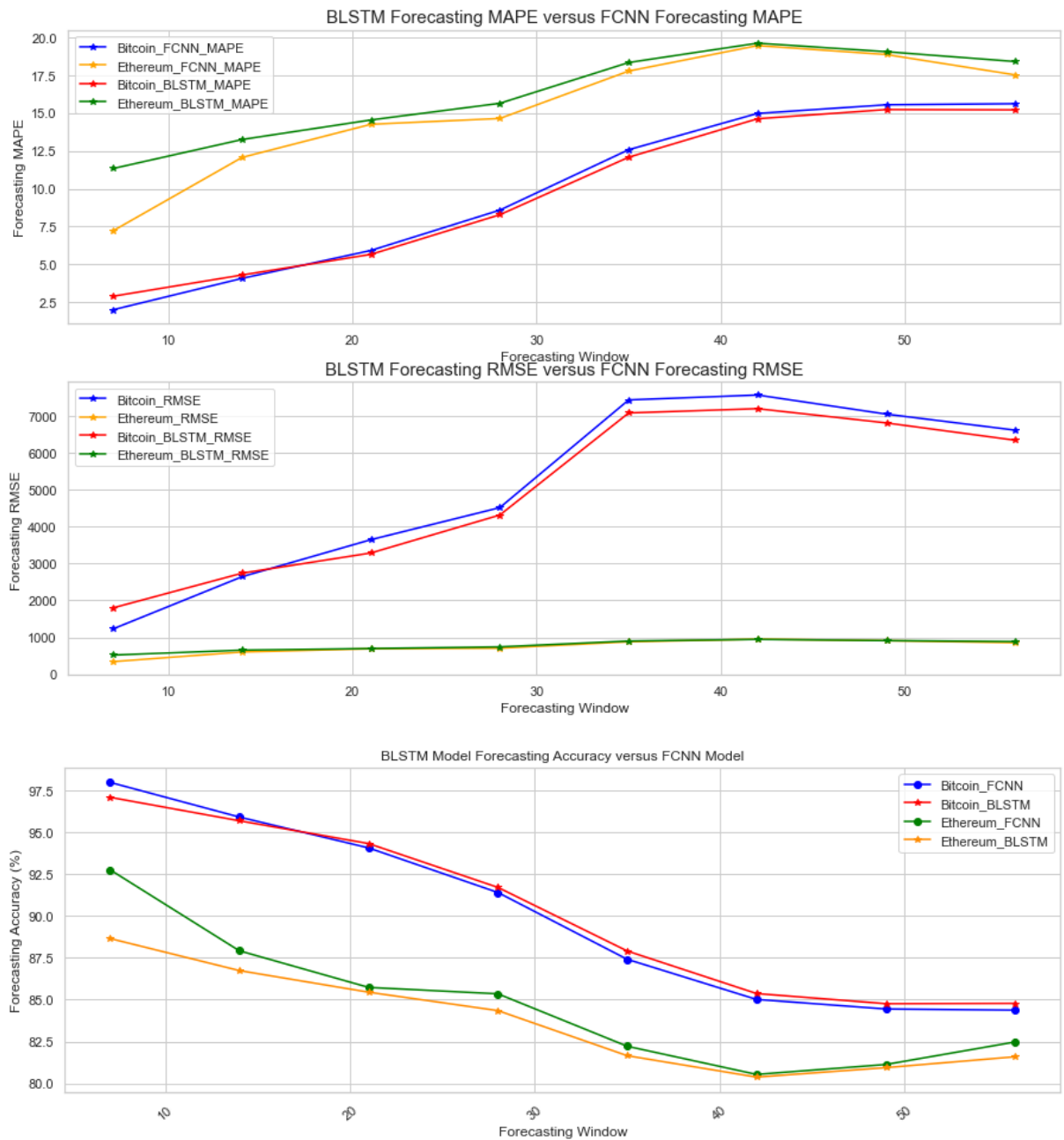


Figure 4.17 BLSTM Model Forecasting Metrics Compared FCNN

4.6 CNN Model Performance

Figure 4.18 (a) and (b) depict the CNN model training and validation results on Bitcoin and Ether close price dataset. The results shown in these figures are for a forecasting window of 7 days. Appendix A shows the rest of the forecasting windows used.

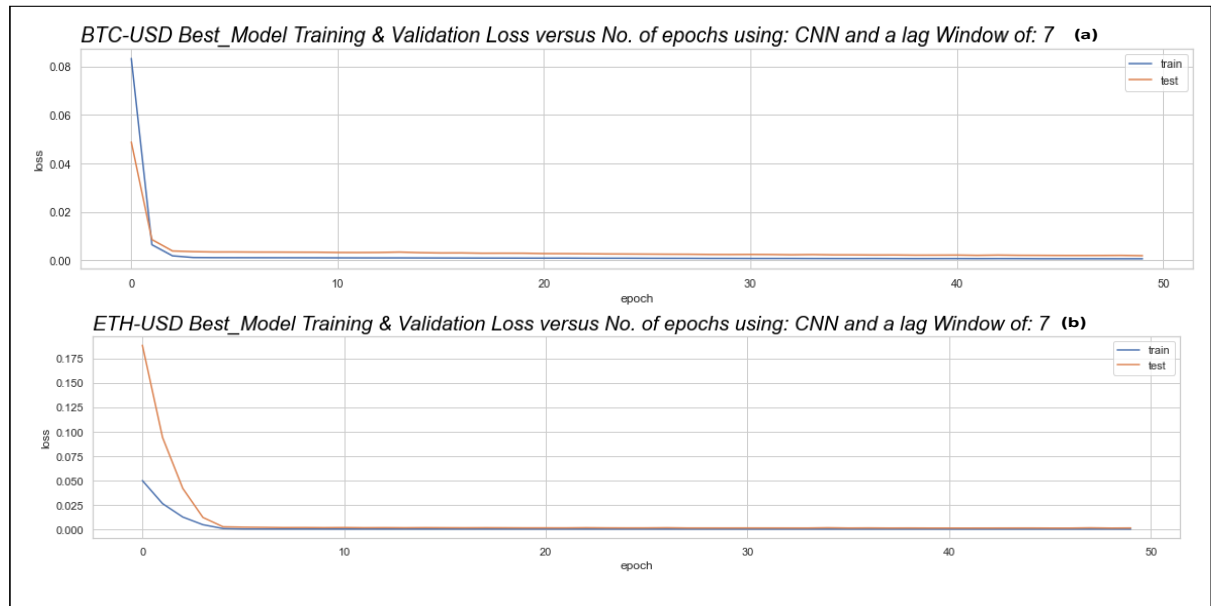


Figure 4.18 CNN Model Fitting and Validation Performance

Figures 4.19(a) and (b) show the best CNN model prediction performance versus the actual values for Bitcoin and Ether, respectively. A complete list of graphs for this model's performance on other forecasting windows can be found in Appendix A.

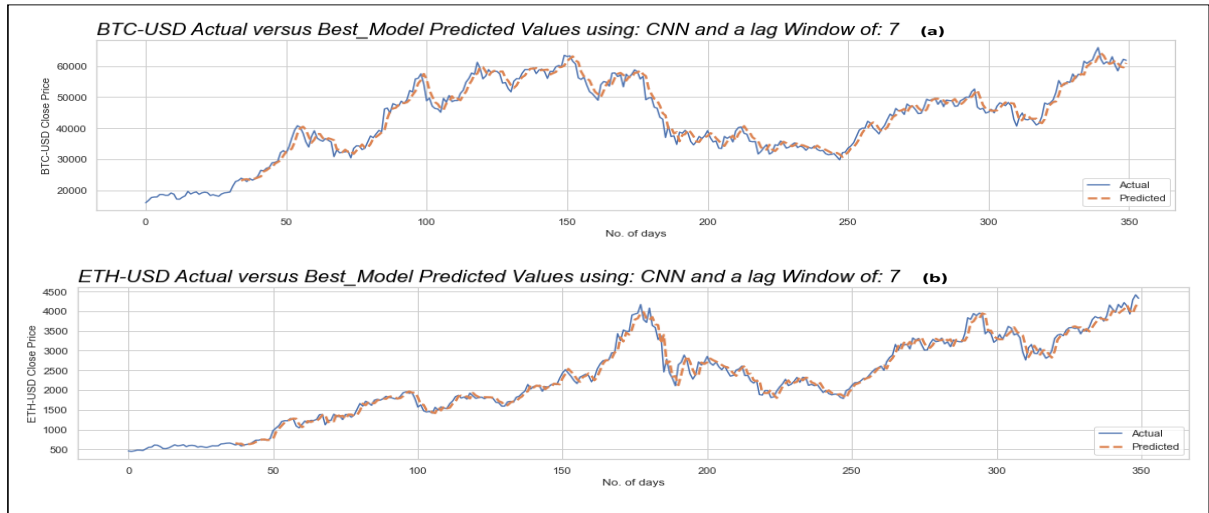


Figure 4.19 CNN Model Prediction Performance

Table 4.6 displays the CNN model forecasting performance metrics for each cryptocurrency, compared to the base FCNN model.

Table 4.6 CNN Model Forecasting Performance

Ticker	Forecasting Window	Forecasting MAE		Forecasting MAPE		Forecasting RMSE		Forecasting Accuracy %	
		FCNN	CNN	FCNN	CNN	FCNN	CNN	FCNN	CNN
BTC-USD	7	1034.86	1085.03	1.99	2.01	1230.99	1265.32	98.01	97.99
	14	2175.30	1940.68	4.06	3.67	2644.16	2403.41	95.94	96.33
	21	2895.15	2573.15	5.91	5.48	3648.56	3217.47	94.09	94.52
	28	3812.29	3781.15	8.58	8.52	4517.06	4479.12	91.42	91.48
	35	5762.85	5712.15	12.58	12.50	7437.53	7386.31	87.42	87.50
	42	6098.32	5949.97	14.98	15.09	7569.70	7561.78	85.02	84.91
	49	5306.14	5200.60	15.55	15.52	7051.97	6957.50	84.45	84.48
	56	4923.56	4870.96	15.63	15.53	6616.87	6601.35	84.37	84.47
ETH-USD	7	325.91	342.16	7.22	7.60	342.04	360.79	92.78	92.40
	14	557.23	564.33	12.07	12.22	602.68	610.68	87.93	87.78
	21	647.84	612.06	14.26	13.51	689.44	649.80	85.74	86.49
	28	659.16	670.11	14.65	14.84	705.03	711.02	85.35	85.16
	35	792.36	803.00	17.78	18.01	879.17	888.83	82.22	81.99
	42	859.62	836.84	19.47	18.94	951.30	928.36	80.53	81.06
	49	820.99	800.46	18.88	18.41	911.49	889.68	81.12	81.59
	56	749.83	760.47	17.53	17.76	853.08	862.84	82.47	82.24

Figure 4.20 shows the forecasting results for seven days forecasting windows against the actual close price of both currencies. The actual close prices are being used to compare the model's accuracy in forecasting future prices.

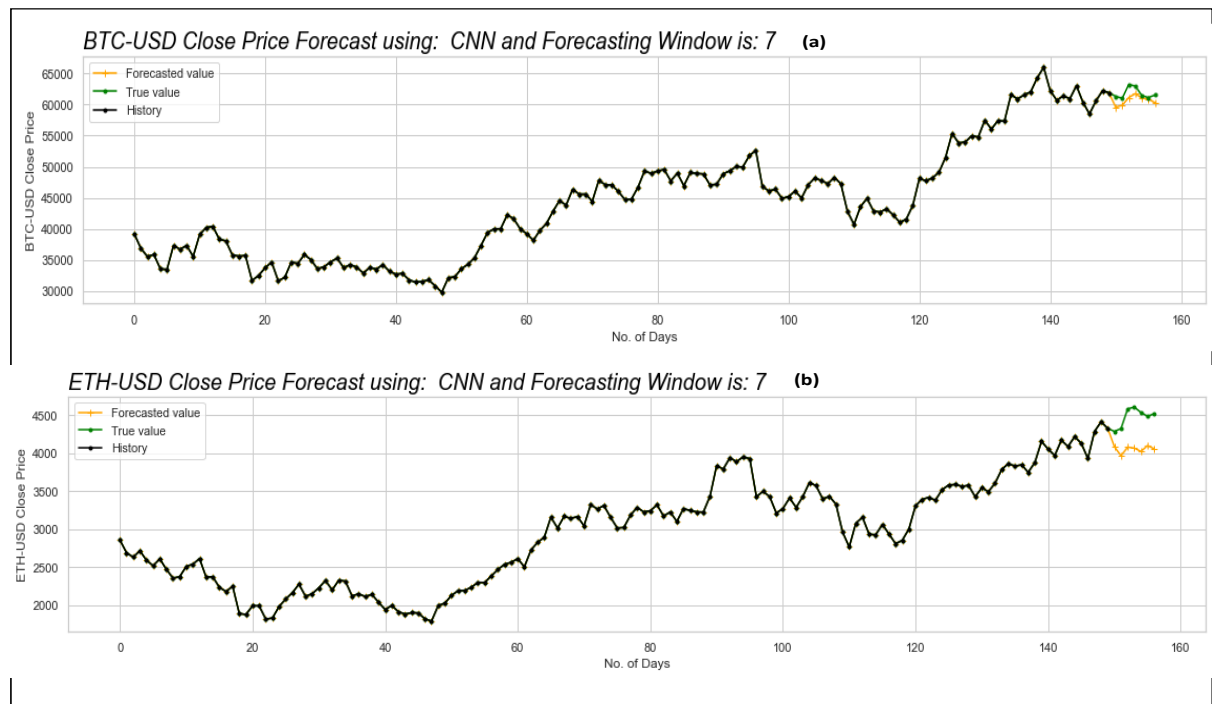


Figure 4.20 CNN Model Forecasting Performance

CNN Model Forecasting Performance

Figure 4.21 depicts the CNN model forecasting performance metrics compared to the FCNN base model performance. For Bitcoin, with an average MAPE of 9.79 across all forecasting windows, the CNN slightly exceeded the FCNN model average MAPE of 9.91. CNN achieved an Accuracy of 90.21% compared to FCNN Accuracy of 90.09%. For Ether, CNN produced the best results with MAPE of 15.11 and Accuracy of 84.84% compared to all models and not just the base model.

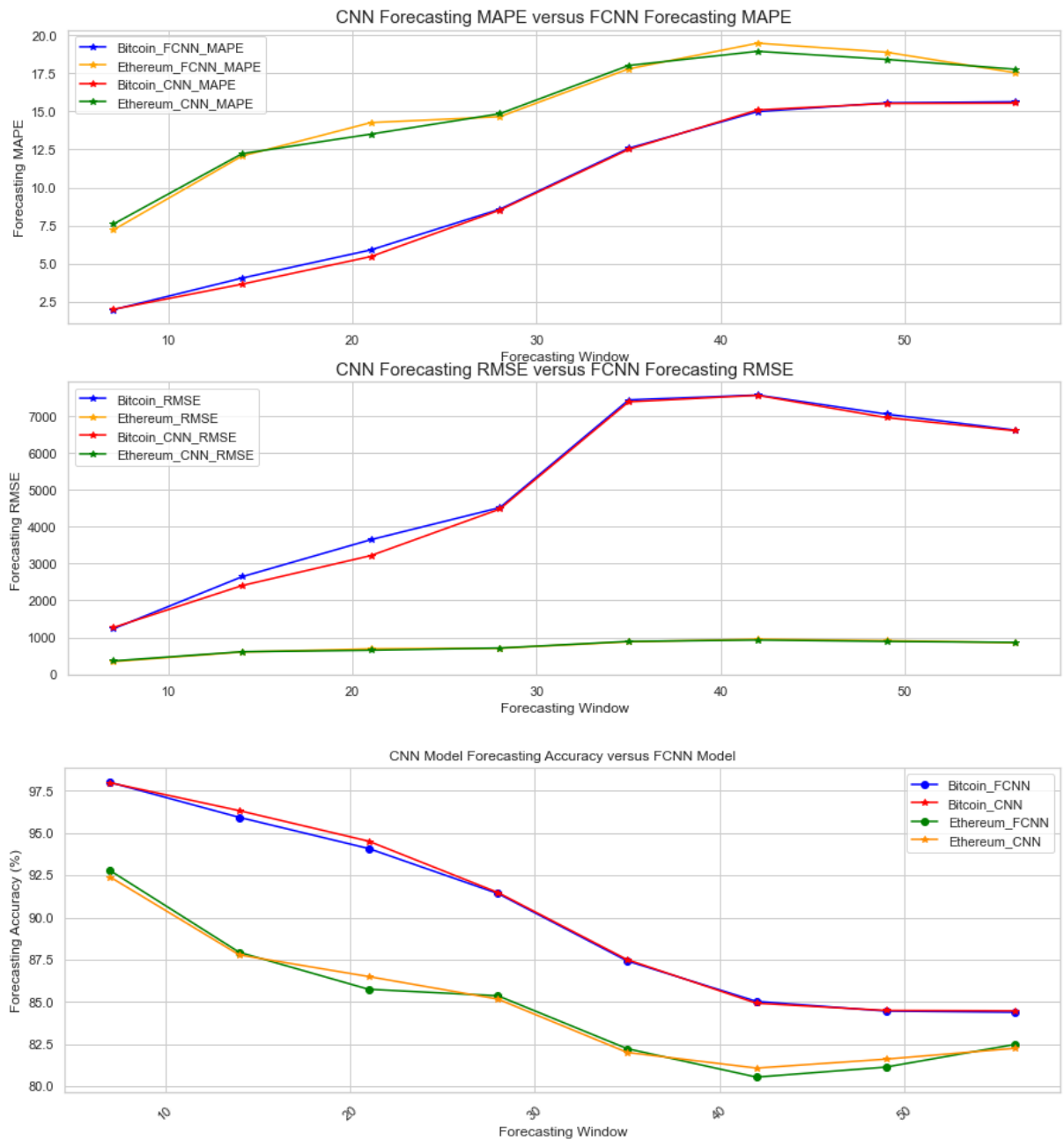


Figure 4.21 CNN Model Performance Metrics Compared to FCNN

4.7 Combined Model 1 (CNN_LSTM_FCNN)

Figure 4.22 (a) and (b) depict the combined model one training and validation results on Bitcoin and Ether close price dataset. The results shown in these figures are for a forecasting window of 7 days. Appendix A provides more information on the rest of the sliding windows used.

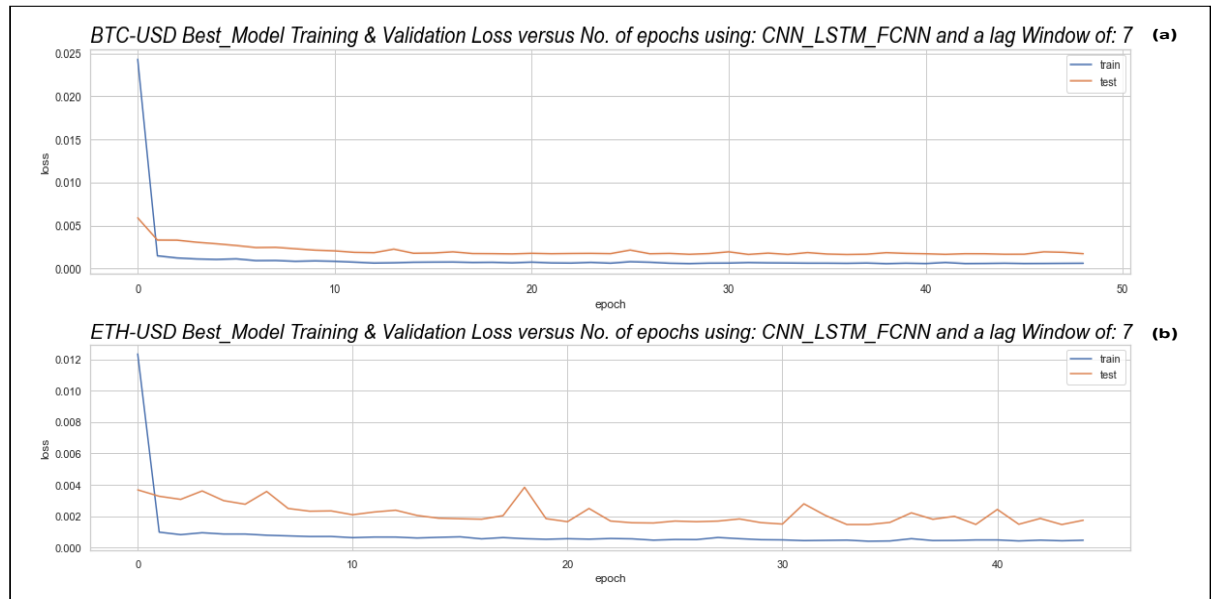


Figure 4.22 Combined Model 1 Fitting and Validation Performance

Figures 4.23(a) and (b) show the best combined model one prediction performance versus the actual values for Bitcoin and Ether, respectively. A complete list of graphs for this model's performance on other forecasting windows are provided in Appendix A.

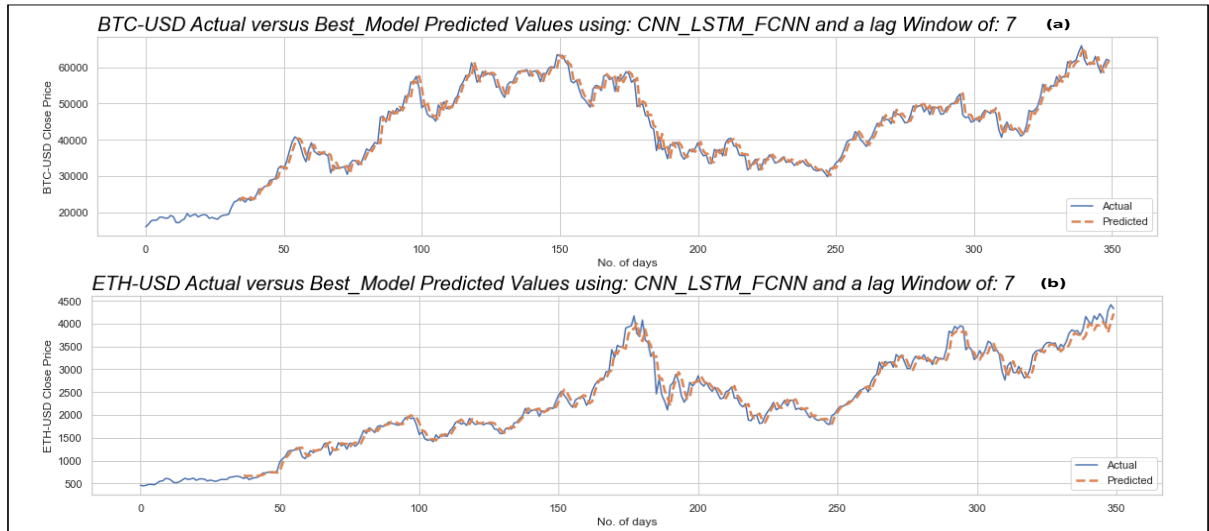


Figure 4.23 Combined Model 1 Prediction Performance

Table 4.7 displays the CNN model forecasting performance metrics for each cryptocurrency and forecasting window compared to the base FCNN model.

Table 4.7 Combined Model 1 (CNN_LSTM_FCNN) Performance

Ticker	Forecasting Window	Forecasting MAE		Forecasting MAPE		Forecasting RMSE		Forecasting Accuracy	
		FCNN	CNN LSTM FCNN	FCNN	CNN LSTM FCNN	FCNN	CNN LSTM FCNN	FCNN	CNN LSTM FCNN
BTC-USD	7	1034.86	1331.11	1.99	2.51	1230.99	1666.07	98.01	97.49
	14	2175.30	2139.68	4.06	4.10	2644.16	2476.33	95.94	95.90
	21	2895.15	2888.28	5.91	5.90	3648.56	3512.84	94.09	94.10
	28	3812.29	3682.99	8.58	8.27	4517.06	4276.92	91.42	91.73
	35	5762.85	5582.68	12.58	11.75	7437.53	6812.13	87.42	88.25
	42	6098.32	5755.57	14.98	14.34	7569.70	7213.93	85.02	85.66
	49	5306.14	5452.61	15.55	15.21	7051.97	7116.69	84.45	84.79
	56	4923.56	4724.49	15.63	14.97	6616.87	6407.44	84.37	85.03
ETH-USD	7	325.91	431.56	7.22	9.58	342.04	450.61	92.78	90.42
	14	557.23	628.54	12.07	13.62	602.68	670.23	87.93	86.38
	21	647.84	668.39	14.26	14.73	689.44	705.67	85.74	85.27
	28	659.16	699.09	14.65	15.57	705.03	734.70	85.35	84.43
	35	792.36	812.17	17.78	18.17	879.17	877.31	82.22	81.83
	42	859.62	889.30	19.47	20.09	951.30	964.41	80.53	79.91
	49	820.99	828.01	18.88	18.97	911.49	916.64	81.12	81.03
	56	749.83	779.05	17.53	18.10	853.08	872.66	82.47	81.90

Figure 4.24 shows the forecasting results for seven-day sliding windows against both currencies' actual close prices. The actual close prices are being used to compare the model's accuracy in forecasting future prices.

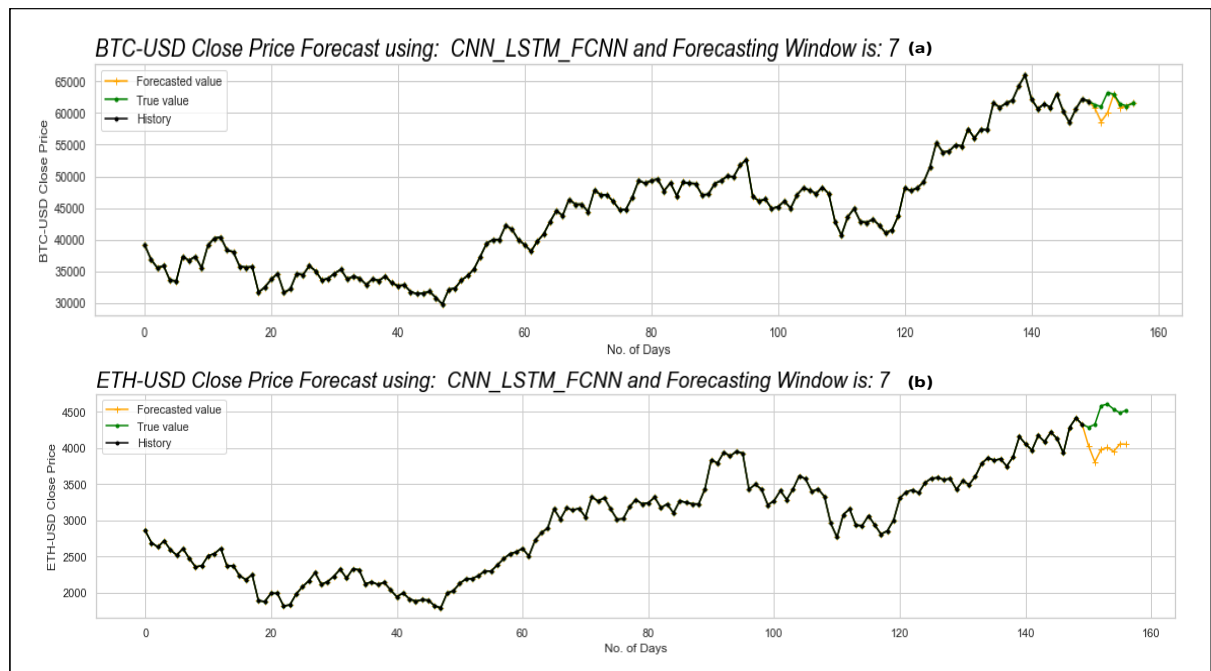


Figure 4.24 Combined Model 1 Forecasting Performance

Figure 4.25 depicts the combined model one forecasting performance metrics results compared to FCNN base model performance. For Bitcoin, this model achieved the best results among all other models with an average MAPE of 9.63 and Accuracy of 90.37%. However, the MAPE score started to increase as the forecasting window increased. The best results were achieved when the forecasting window was equal to 7 days. For Ether, this model underperformed when compared to the base model. It achieved an average MAPE of 16.59, which is higher than the base model score of 15.22 for MAPE, and an Accuracy of 83.90%, which is lower than the base model score of 84.77%.

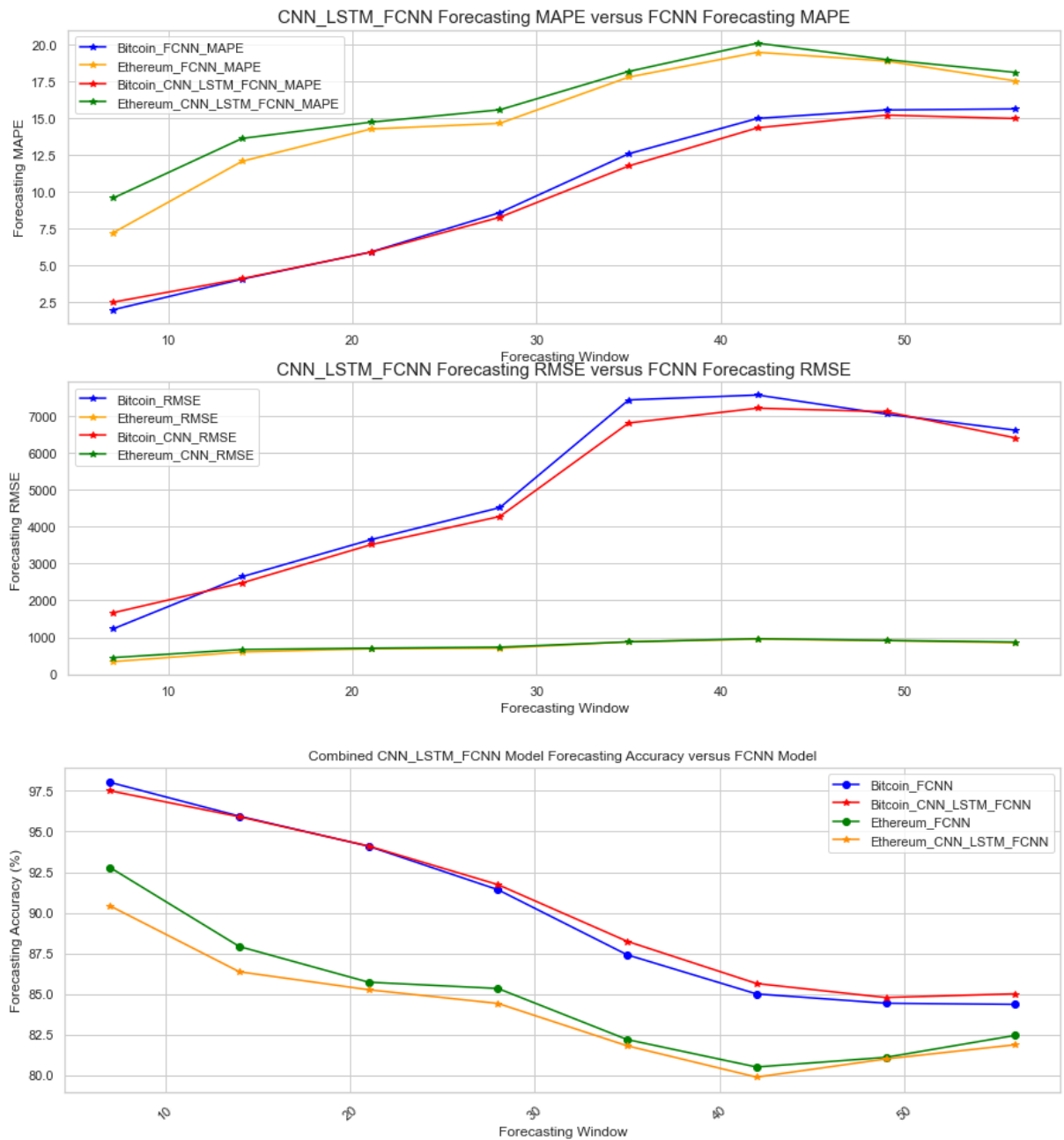


Figure 4.25 Combined Model 1 Performance Compared to FCNN

4.8 Combined Model 2 (CNN_BLSTM_FCNN)

Figure 4.26 (a) and (b) depict the combined model two training and validation results on Bitcoin Ether close price dataset. The results shown in these figures are for a forecasting window of 7 days. Appendix A provides more information on the rest of the forecasting windows used.

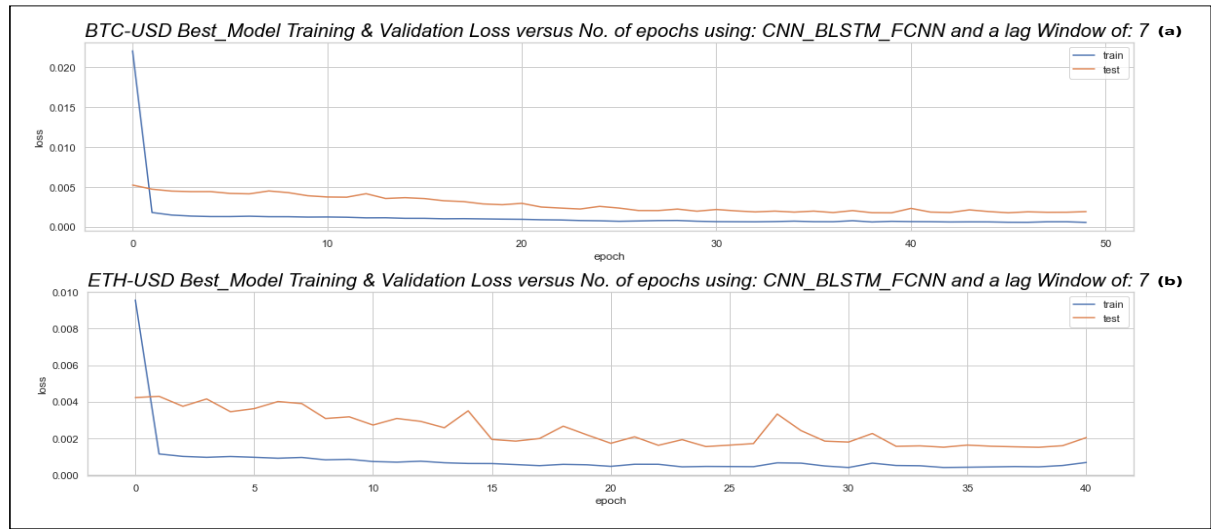


Figure 4.26 Combined Model 2 Fitting and Validation Performance

Figures 4.27(a) and (b) show the best combined model one prediction performance versus the actual values for Bitcoin and Ether, respectively. A complete list of graphs for this model's performance on other forecasting windows are provided in Appendix A.

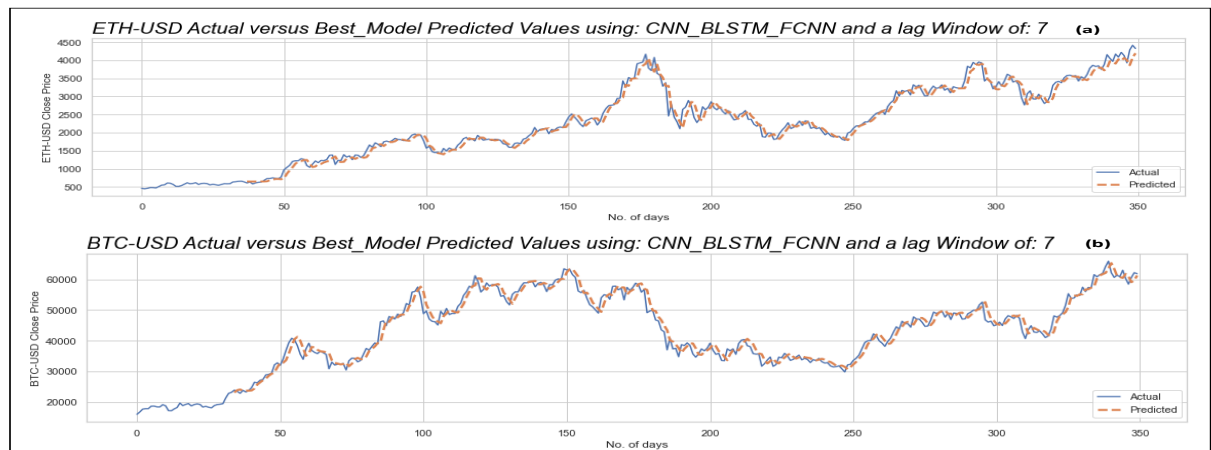


Figure 4.27 Combined Model 2 Prediction Performance

Table 4.8 displays the combined model two forecasting performance metrics for each cryptocurrency and forecasting window compared to the base FCNN model.

Table 4.8 Combined Model 2 (CNN_BLSTM_FCNN) Performance

Ticker	Forecasting Window	Forecasting MAE		Forecasting MAPE		Forecasting RMSE		Forecasting Accuracy %	
		FCNN	CNN BLSTM FCNN	FCNN	CNN BLSTM FCNN	FCNN	CNN BLSTM FCNN	FCNN	CNN BLSTM FCNN
BTC-USD	7	1034.86	1210.91	1.99	2.33	1230.99	1519.93	98.01	97.67
	14	2175.30	2101.01	4.06	4.13	2644.16	2455.50	95.94	95.87
	21	2895.15	2760.12	5.91	5.85	3648.56	3324.56	94.09	94.15
	28	3812.29	3637.54	8.58	8.51	4517.06	4428.63	91.42	91.49
	35	5762.85	5670.07	12.58	12.33	7437.53	7293.50	87.42	87.67
	42	6098.32	5733.97	14.98	14.81	7569.70	7308.62	85.02	85.19
	49	5306.14	5218.69	15.55	15.30	7051.97	6884.34	84.45	84.70
	56	4923.56	4645.26	15.63	15.32	6616.87	6399.15	84.37	84.68
ETH-USD	7	325.91	392.91	7.22	8.71	342.04	417.14	92.78	91.29
	14	557.23	601.83	12.07	13.04	602.68	649.99	87.93	86.96
	21	647.84	640.47	14.26	14.15	689.44	681.13	85.74	85.85
	28	659.16	689.43	14.65	15.28	705.03	727.49	85.35	84.72
	35	792.36	817.21	17.78	18.24	879.17	903.89	82.22	81.76
	42	859.62	871.27	19.47	19.65	951.30	951.67	80.53	80.35
	49	820.99	833.60	18.88	19.07	911.49	915.47	81.12	80.93
	56	749.83	783.31	17.53	18.19	853.08	878.32	82.47	81.81

Figure 4.28 shows the forecasting results for seven days sliding windows against the actual close price of both currencies, which were downloaded as soon they were published on the Yahoo Finance site. The actual close prices are being used to compare how accurately the model forecasts future prices.

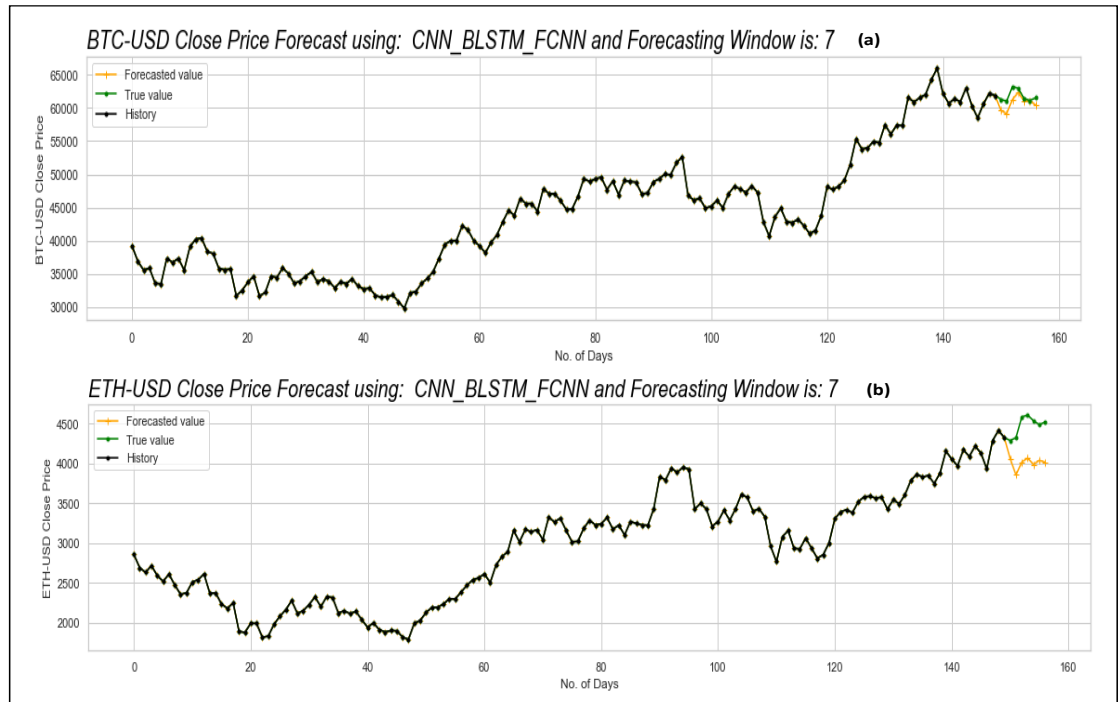


Figure 4.28 Combined Model 2 Forecasting Performance

The Combined Model 2 results compared to the FCNN base model performance are shown in Figure 4.29. The results show that this model scored a MAPE of 9.82 and an Accuracy of 90.18% for Bitcoin. Moreover, it scored a MAPE of 16.82 and an Accuracy of 84.21% for Ether. For both cryptocurrencies, the model accuracy declined as the forecasting window increased. The best Accuracy of 91.29% was achieved when the forecasting window was seven days.

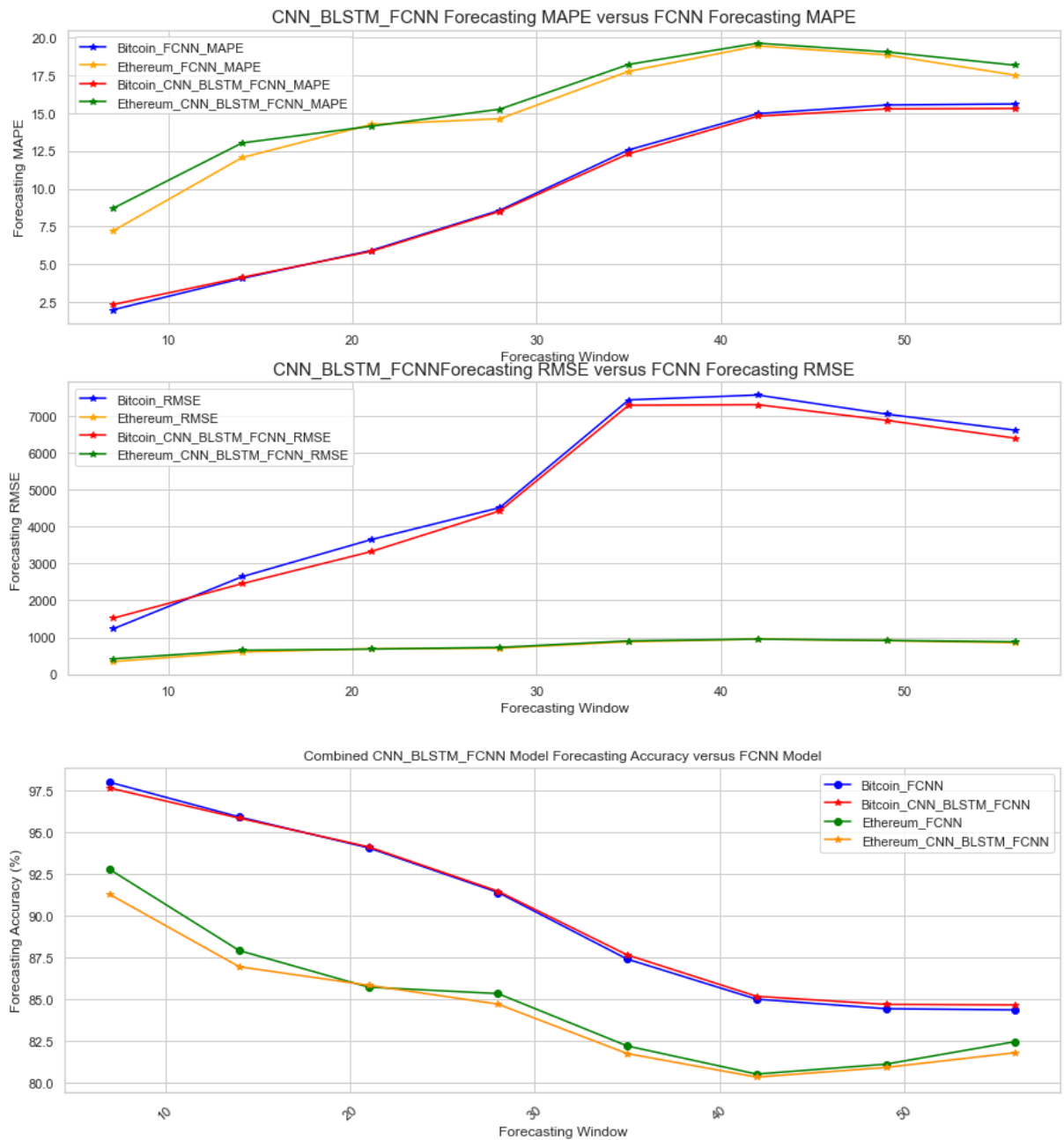


Figure 4.29 Combined Model 2 Performance Compared to FCNN

4.9 Overall Models Performance Results

This section presents the results obtained from the experiments in terms of the measures MAE, MAPE, RMSE, and Accuracy for all of the six models for the Bitcoin and Ether datasets compared with the FCNN baseline model. Table 4.9 summarizes the average performance results for applying the six models on the Bitcoin dataset using different forecasting horizons. Figure 4.30 presents a graphical comparison between the six models using the model's performance evaluation metrics. Comparing all models, Combined Model 1 scored the lowest at 9.63 for MAPE. Table 4.10 presents the model's performance results for the Bitcoin dataset for all forecasting horizons.

Table 4.9 Models' Average Performance for Bitcoin

Model Name	MAE	MAPE	RMSE	Forecasting Accuracy %
BLSTM	3939.36	9.78	4948.22	90.22
CNN	3889.21	9.79	4984.03	90.21
CNN_BLSTM_FCNN	3872.20	9.82	4951.78	90.18
CNN_LSTM_FCNN	3944.68	9.63	4935.29	90.37
FCNN	4001.06	9.91	5089.61	90.09
LSTM	4111.05	9.65	5136.47	90.35

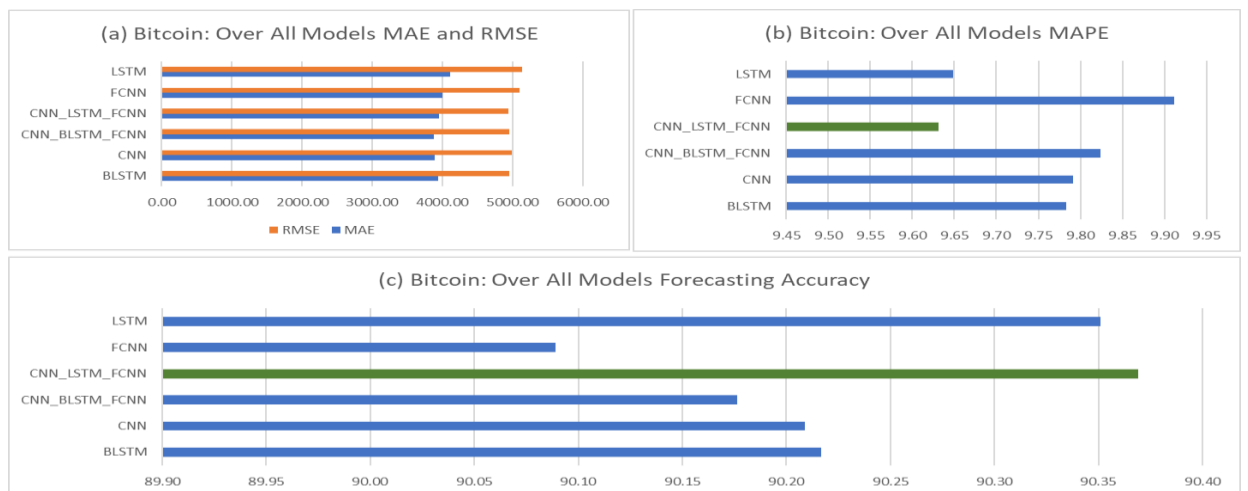


Figure 4.30 Models' Average Performance for Bitcoin

Table 4.10 Detailed Models' Performance for Bitcoin

ticker	forecasting windows	model name	mae	mape	rmse	Forecasting accuracy %
BTC-USD	7	BLSTM	1744.03	2.88	1800.02	97.12
		CNN	1085.03	2.01	1265.32	97.99
		CNN_BLSTM_FCNN	1210.91	2.33	1519.93	97.67
		CNN_LSTM_FCNN	1331.11	2.51	1666.07	97.49
		FCNN	1034.86	1.99	1230.99	98.01
		LSTM	1197.11	2.03	1316.08	97.97
	14	BLSTM	2333.58	4.29	2739.98	95.71
		CNN	1940.68	3.67	2403.41	96.33
		CNN_BLSTM_FCNN	2101.01	4.13	2455.50	95.87
		CNN_LSTM_FCNN	2139.68	4.10	2476.33	95.90
		FCNN	2175.30	4.06	2644.16	95.94
		LSTM	2270.74	3.86	2789.11	96.14
	21	BLSTM	2633.73	5.65	3287.77	94.35
		CNN	2573.15	5.48	3217.47	94.52
		CNN_BLSTM_FCNN	2760.12	5.85	3324.56	94.15
		CNN_LSTM_FCNN	2888.28	5.90	3512.84	94.10
		FCNN	2895.15	5.91	3648.56	94.09
		LSTM	2822.10	5.56	3697.49	94.44
	28	BLSTM	3587.49	8.28	4316.28	91.72
		CNN	3781.15	8.52	4479.12	91.48
		CNN_BLSTM_FCNN	3637.54	8.51	4428.63	91.49
		CNN_LSTM_FCNN	3682.99	8.27	4276.92	91.73
		FCNN	3812.29	8.58	4517.06	91.42
		LSTM	3837.81	8.19	4441.06	91.81
	35	BLSTM	5683.72	12.08	7084.23	87.92
		CNN	5712.15	12.50	7386.31	87.50
		CNN_BLSTM_FCNN	5670.07	12.33	7293.50	87.67
		CNN_LSTM_FCNN	5582.68	11.75	6812.13	88.25
		FCNN	5762.85	12.58	7437.53	87.42
		LSTM	5831.41	12.04	7245.29	87.96
	42	BLSTM	5710.70	14.63	7199.81	85.37
		CNN	5949.97	15.09	7561.78	84.91
		CNN_BLSTM_FCNN	5733.97	14.81	7308.62	85.19
		CNN_LSTM_FCNN	5755.57	14.34	7213.93	85.66
		FCNN	6098.32	14.98	7569.70	85.02
		LSTM	6339.55	14.81	7784.86	85.19
	49	BLSTM	5238.35	15.24	6813.63	84.76
		CNN	5200.60	15.52	6957.50	84.48
		CNN_BLSTM_FCNN	5218.69	15.30	6884.34	84.70
		CNN_LSTM_FCNN	5452.61	15.21	7116.69	84.79
		FCNN	5306.14	15.55	7051.97	84.45
		LSTM	5561.13	15.35	7186.71	84.65
	56	BLSTM	4583.28	15.22	6344.06	84.78
		CNN	4870.96	15.53	6601.35	84.47
		CNN_BLSTM_FCNN	4645.26	15.32	6399.15	84.68
		CNN_LSTM_FCNN	4724.49	14.97	6407.44	85.03
		FCNN	4923.56	15.63	6616.87	84.37
		LSTM	5028.58	15.36	6631.13	84.64

Table 4.11 summarizes the average performance results for applying the six models on the Ether dataset using different forecasting windows. Figure 4.31 presents a graphical comparison between the six models using the model's performance evaluation metrics. The CNN model scored 15.16 for MAPE, and the FCNN scored a MAPE of 15.23. These scores are the lowest compared to all models for the Ether cryptocurrency.

Table 4.11 Models' Average Performance for Ether

Model Name	MAE	MAPE	RMSE	Forecasting Accuracy %
BLSTM	724.99	16.28	781.66	83.72
CNN	673.68	15.16	737.75	84.84
CNN_BLSTM_FCNN	703.75	15.79	765.64	84.21
CNN_LSTM_FCNN	717.02	16.10	774.03	83.90
FCNN	676.62	15.23	741.78	84.77
LSTM	741.61	16.64	794.37	83.36

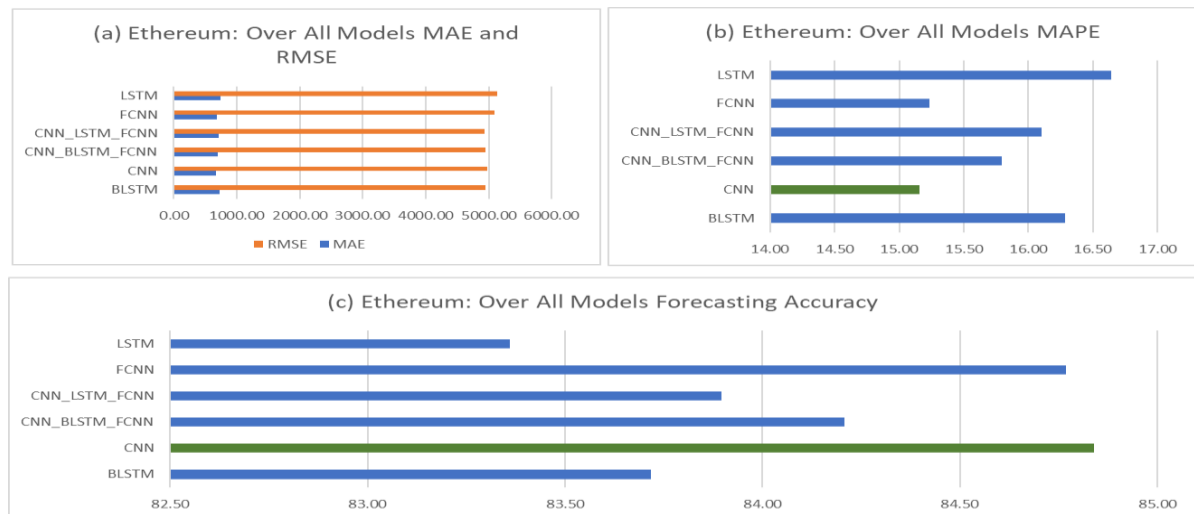


Figure 4.31 Models' Average Performance for Ether

Table 4.13 presents models performance results for the Ether dataset for all forecasting horizons.

Table 4.12 Detailed Models' Performance for Ether

ticker	forecasting windows	model name	mae	mape	rmse	forecasting accuracy
Ether	7	BLSTM	510.39	11.34	520.18	88.66
		CNN	342.16	7.60	360.79	92.40
		CNN_BLSTM_FCNN	392.91	8.71	417.14	91.29
		CNN_LSTM_FCNN	431.56	9.58	450.61	90.42
		FCNN	325.91	7.22	342.04	92.78
		LSTM	434.11	9.63	447.00	90.37
	14	BLSTM	611.75	13.25	653.11	86.75
		CNN	564.33	12.22	610.68	87.78
		CNN_BLSTM_FCNN	601.83	13.04	649.99	86.96
		CNN_LSTM_FCNN	628.54	13.62	670.23	86.38
		FCNN	557.23	12.07	602.68	87.93
		LSTM	685.97	14.87	716.89	85.13
	21	BLSTM	661.50	14.55	695.98	85.45
		CNN	612.06	13.51	649.80	86.49
		CNN_BLSTM_FCNN	640.47	14.15	681.13	85.85
		CNN_LSTM_FCNN	668.39	14.73	705.67	85.27
		FCNN	647.84	14.26	689.44	85.74
		LSTM	695.24	15.28	733.71	84.72
	28	BLSTM	702.04	15.65	742.57	84.35
		CNN	670.11	14.84	711.02	85.16
		CNN_BLSTM_FCNN	689.43	15.28	727.49	84.72
		CNN_LSTM_FCNN	699.09	15.57	734.70	84.43
		FCNN	659.16	14.65	705.03	85.35
		LSTM	722.17	16.10	759.12	83.90
	35	BLSTM	818.64	18.35	899.68	81.65
		CNN	803.00	18.01	888.83	81.99
		CNN_BLSTM_FCNN	817.21	18.24	903.89	81.76
		CNN_LSTM_FCNN	812.17	18.17	877.31	81.83
		FCNN	792.36	17.78	879.17	82.22
		LSTM	853.48	19.11	918.85	80.89
	42	BLSTM	868.03	19.63	944.84	80.37
		CNN	836.84	18.94	928.36	81.06
		CNN_BLSTM_FCNN	871.27	19.65	951.67	80.35
		CNN_LSTM_FCNN	889.30	20.09	964.41	79.91
		FCNN	859.62	19.47	951.30	80.53
		LSTM	887.60	20.06	960.00	79.94
	49	BLSTM	832.04	19.07	911.74	80.93
		CNN	800.46	18.41	889.68	81.59
		CNN_BLSTM_FCNN	833.60	19.07	915.47	80.93
		CNN_LSTM_FCNN	828.01	18.97	916.64	81.03
		FCNN	820.99	18.88	911.49	81.12
		LSTM	847.37	19.44	928.58	80.56
	56	BLSTM	795.54	18.41	885.18	81.59
		CNN	760.47	17.76	862.84	82.24
		CNN_BLSTM_FCNN	783.31	18.19	878.32	81.81
		CNN_LSTM_FCNN	779.05	18.10	872.66	81.90
		FCNN	749.83	17.53	853.08	82.47
		LSTM	806.90	18.63	890.78	81.37

Error terms have the same unit of measurement as data units, so the Mean Absolute Error will depend on scale. Thus, this metric cannot be used for comparing times series of different scale units. Figure 4.32 presents a graphical comparison for the MAE of the six models.

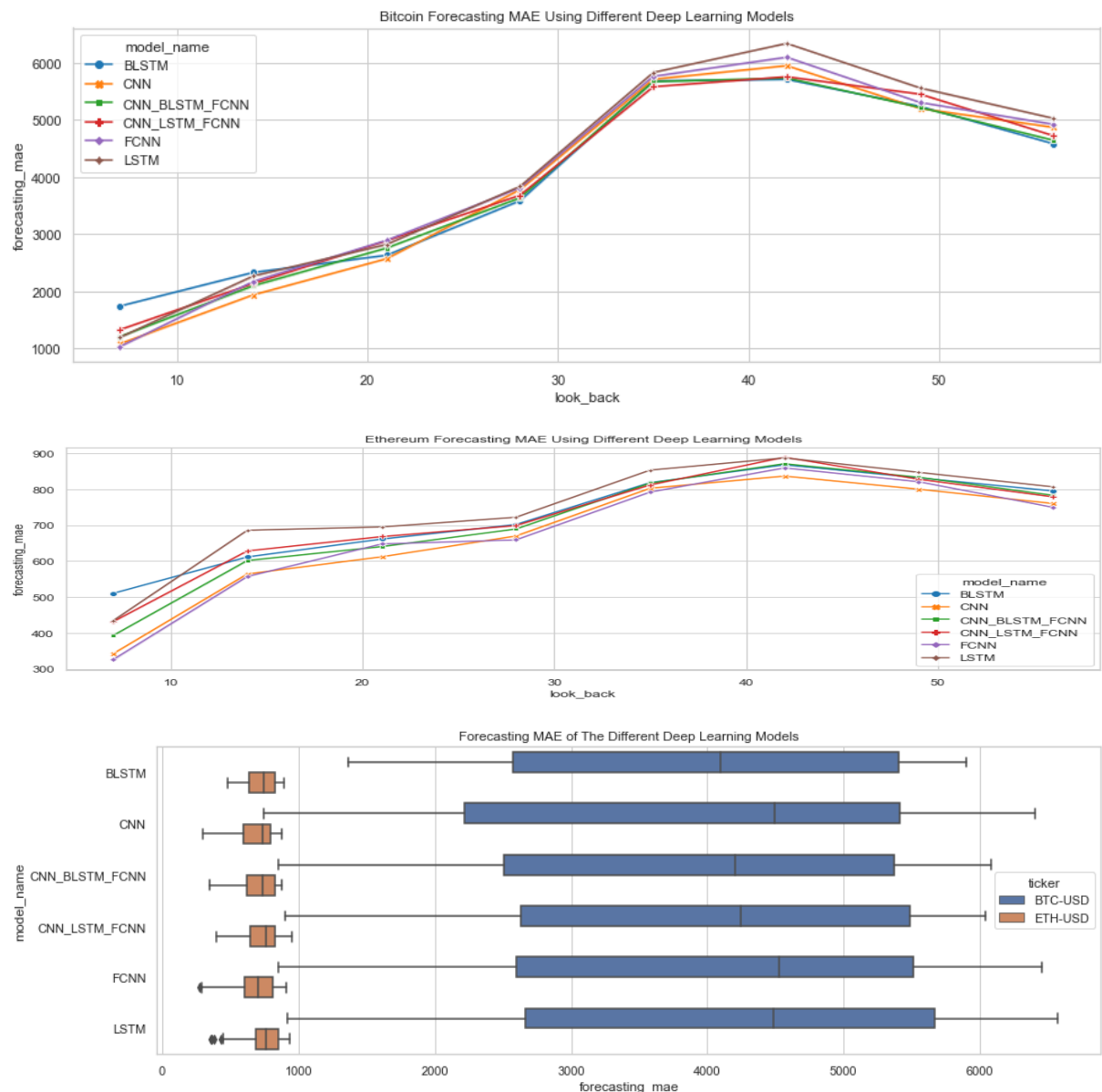


Figure 4.32 Comparison of All Models' Forecasting MAE

A forecast model's accuracy is measured by the mean absolute percentage error (MAPE). This Accuracy is measured as a percentage and can be calculated as the average absolute percent error for each predicted value minus the corresponding actual value divided by the actual value. The most widely used method for assessing forecast accuracy is the MAPE. Moreover, it is a percentage error that is scale-independent, and it can be applied to compare series that have different scales. In Figure 4.33, six models are compared graphically for their MAPE.

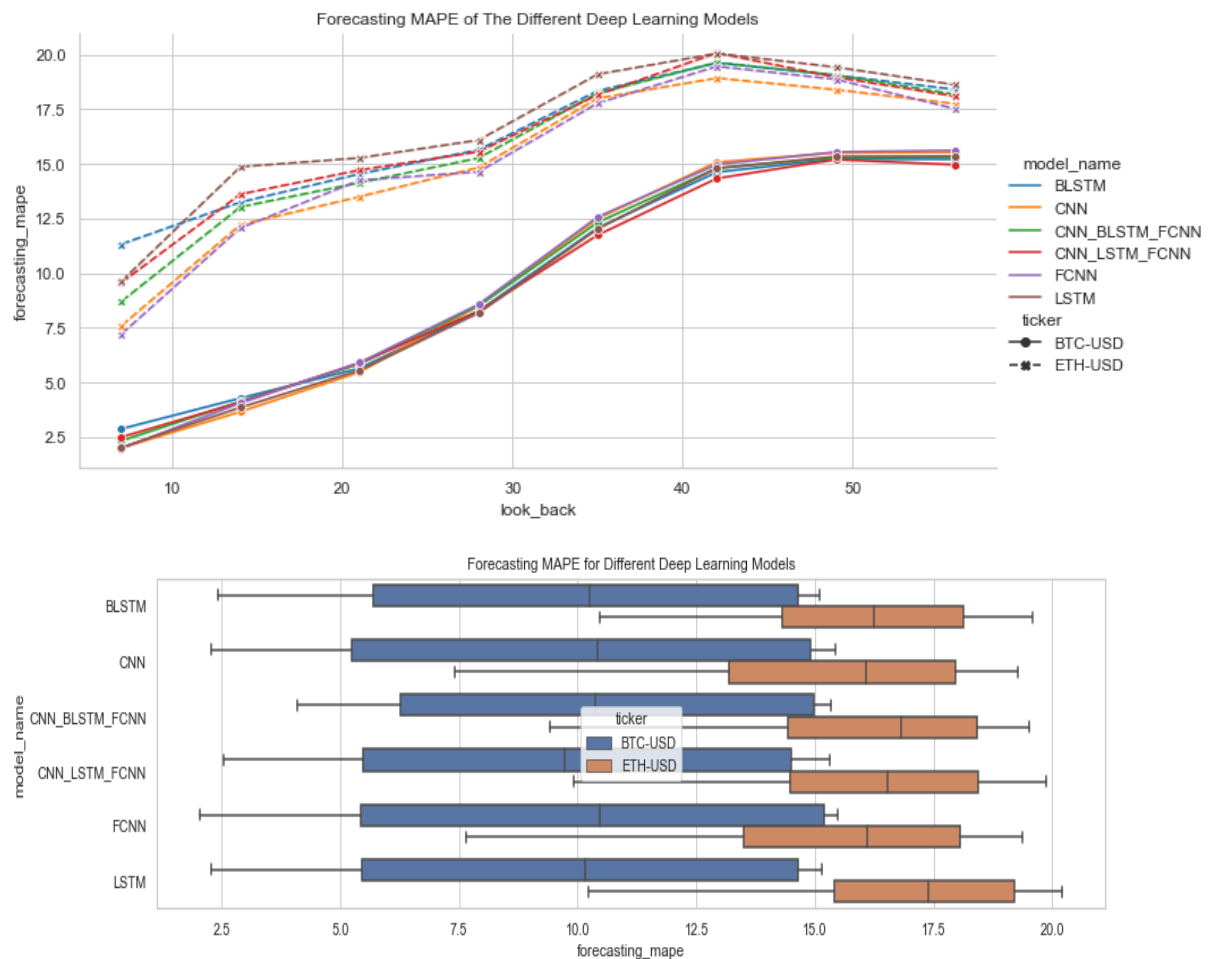


Figure 4.33 Comparison of All Models' Forecasting MAPE

The RMSE is scale-dependent, which means it cannot be compared across series measured at different scales. Additionally, it is measured in the same unit as the data. A graphical representation of the RMSE of the six models is presented in Figure 4.34.

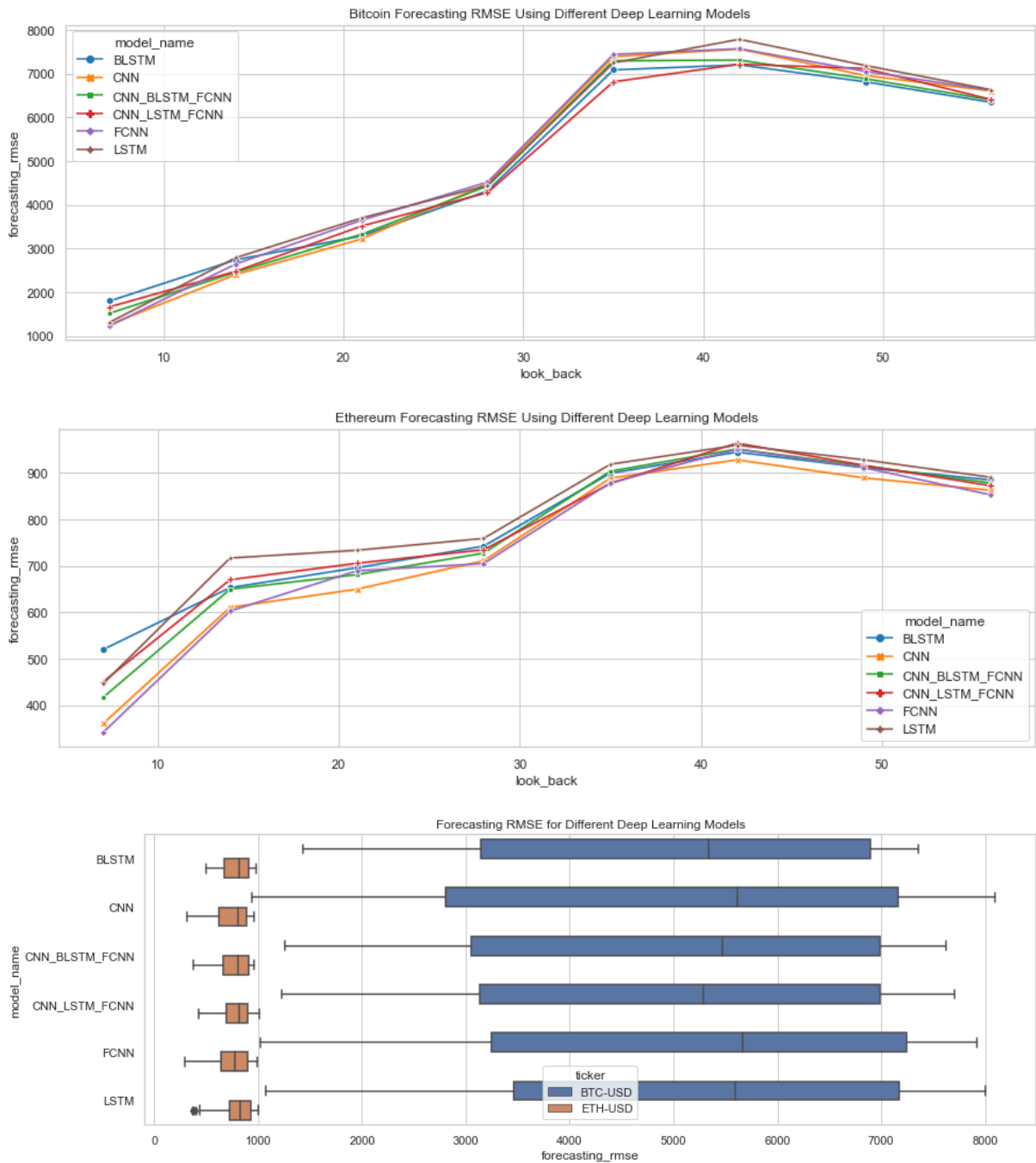


Figure 4.34 Comparison of All Models' Forecasting RMSE

Figure 4.35 presents a graphical comparison for the MAE of the six models. This section provides more information to help answer the question: which DL model is most effective at forecasting cryptocurrency prices, Bitcoin and Ether? When there are no extremes in the data (and no zeros), the mean absolute percentage error (MAPE) is the most commonly used measure to forecast error. In the case of the datasets used for this study, the actual close prices did not include any zero values, so MAPE is safe to be used to measure forecasting accuracy. MAPE was used to calculate the model forecasting accuracy percentage. This calculated metric was used to compare the different models forecasting percentage errors to derive the model accuracy percentage.

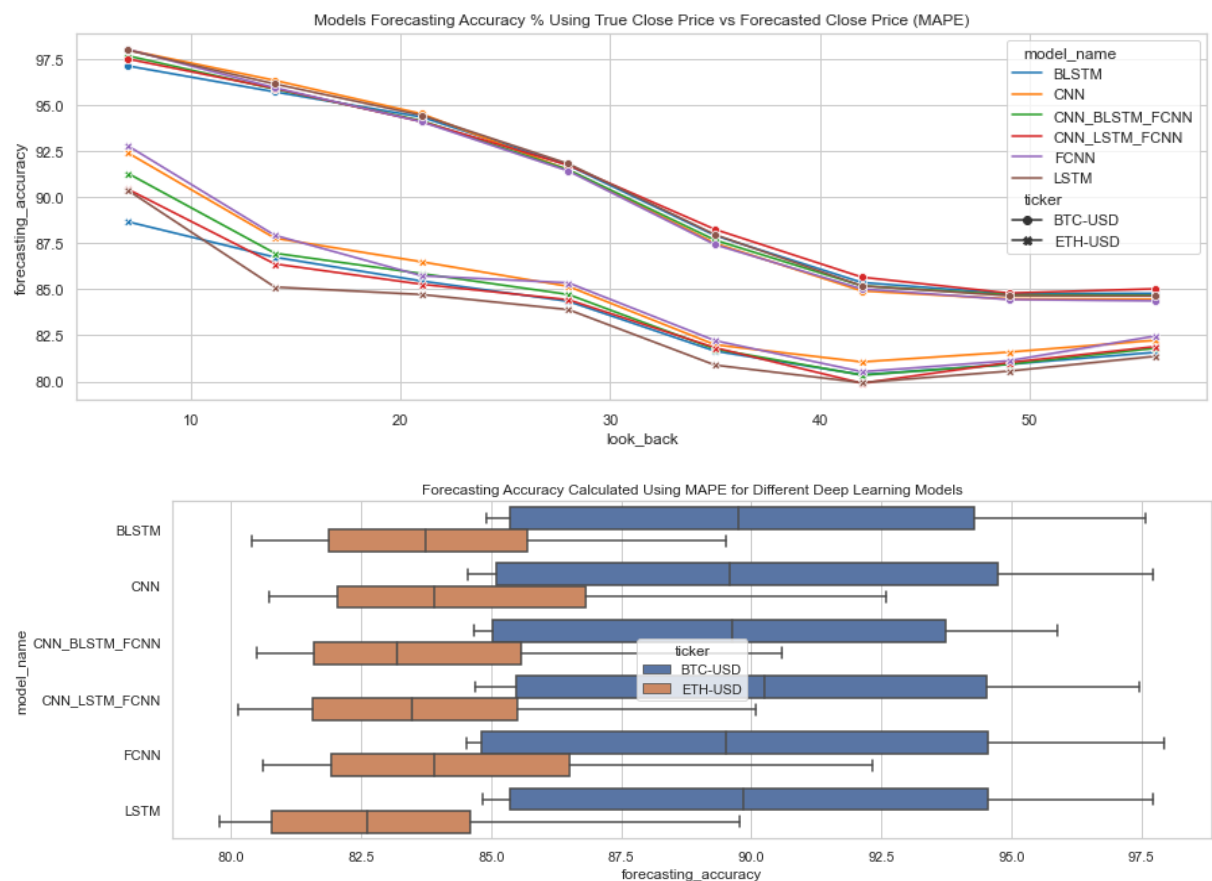


Figure 4.35 Comparison of All Models' Forecasting Accuracy %

Chapter Five:

Discussion, Conclusion and Recommendations

To fill a gap in the state of knowledge regarding the use of deep learning for predicting cryptocurrency prices, a comprehensive experiment was conducted to evaluate and compare the performance of six DL techniques. The forecasting performance of six types of DL techniques was evaluated and compared to a DL base model. These models are FCNN, LSTM, BLSTM, CNN, a combination of CNN, LSTM, followed by FCNN, and one model combined architecture consisted of CNN, BLSTM, and FCNN. The current top two cryptocurrencies, Bitcoin and Ether, datasets were used to evaluate these models.

Combining different DL techniques in an optimized architecture can be an effective and efficient technique for predicting cryptocurrencies' prices. This study investigated a univariate time series forecasting problem with a single-step iterative forecasting method over eight forecasting windows 7, 14, 21, 28, 35, 42, 49, and 56. The results indicate that deep learning techniques are effective in predicting Bitcoin price with an overall average MAPE of 9.76, RMSE of 5007.75, and Accuracy of 90.24% and less effective for Ether, over long-term forecast windows, with an overall average MAPE of 15.85, RMSE of 765.87 and an average accuracy of 84.13%.

Combining CNN, LSTM, and FCNN, model one outperforms the base model and all other models for Bitcoin. Compared to the FCNN base model, the Combined Model 1 scored 2.83% lower MAPE and 3.03% lower RMSE. However, Ether CNN stood up to the base model and outperformed all other models by 0.47% lower MAPE and 0.54% lower

RMSE compared to the base model. Using the FCNN as a base model, especially after optimizing it, to compare DL techniques, rather than against ML or traditional statical methods, is suggested to be more realistic for such studies.

5.1 Study Principles Findings

Cryptocurrency prediction

All models scored satisfactory results for both currencies datasets over a small forecasting horizons such as seven days, but their forecasting accuracy deteriorated over broader forecasting horizons. This is due to the single-step forecasting method's limited capacity than the multi-step method or its variation. MAPE started to increase, and model accuracy decreased when the forecasting windows increased. These findings agree with Taieb and Hyndman (2012), who concluded that the residual error becomes significant with the recursive method.

Using the latest real cryptocurrencies prices to evaluate the accuracy of the forecasted prices was an effective and less costly model deployment simulation technique. The results suggest that DL techniques produced the best forecasting accuracy for cryptocurrency prices over short-term forecast windows if used with the recursive single step forecasting strategy; this is in line with (Khedr et al., 2021).

FCNN as a base model

The FCNN model's results for both cryptocurrencies across different forecast horizons suggest that it achieved satisfactory performance for predicting both cryptocurrencies' close prices for all forecasting windows

using the single-step model. For Bitcoin, the optimized model scored the best value of MAPE 1.99 and RMSE 1230.00 over seven days forecasting window. For Ether, it scored a MAPE of 7.22 and RMSE of 340.04 over the same forecast horizon. In this work, it was noticed that as the forecasting horizons increased in size, the model performance deteriorated. This result was expected due to the residual error accumulation when using the recursive model for forecasting. The results suggest that this model can be used as a baseline to benchmark the other five models' performance.

LSTM model

Table 4.4 shows the results for the LSTM model. The LSTM model outperformed the base model with MAPE 9.65, RMSE 5136.47, and 90.09 Accuracy for Bitcoin. However, it scored lower than the base model for Ether. This model scored an overall average for MAPE 17.66, RMSE 794.37, and 83.36% accurate for Ether. These scores are lower than the base model by 9.24% for MAPE and 7.09% RMSE. The results indicate that LSTM is not the optimal model for Ether and the Ether dataset characteristics and size need to be investigated further. However, LSTM scored the second best model Bitcoin with 0.19% lower than the Combined Model 1. Using Keras Gridsearch library to find the optimal architecture was dropped from this study due to time and processing power constraints on running this model. It took more than 36 hours to run the Gridsearch process for only a single hyperparameters configuration. Instead, the trials and errors technique was used to find the optimal model.

BLSTM model

Table 4.5 shows the results for the BLSTM model. The BLSTM model outperformed the base model with MAPE 9.78, RMSE 4848.22, and 90.22% accuracy for Bitcoin. However, it scored lower than the base model for Ether. This model scored an overall average for MAPE 16.28, 6.89% higher than the base model, and an RMSE of 781.66, 5.3% higher than the base model. Although the results indicate that BLSM is a good DL technique to predict Bitcoin, it underperformed for Ether compared to the base and CNN models. (Singh et al., 2021)

CNN model

With CNN, features can be automatically extracted from raw data without requiring prior knowledge or engineering. The results for the CNN One-dimensional model are shown in Table 4.6. The CNN model outperformed the base model scoring a MAPE 15.16, RMSE 737.75, and 84.84% accurate for Ether. These results indicate that CNN was the optimal technique for Ether and a highly competitive technique for Bitcoin, scoring a MAPE of 9.79 and RMSE of 4984.03 and 90.21% for Accuracy. The results support what is reported by (Koprinska et al., 2018). Based on their analysis of the energy consumption data, they discovered that CNN outperforms MLP at time series forecasting.

Combined Model One (CNN, LSTM and FCNN)

Table 4.7 presents the results for the Combined model 1 (CNN, LSTM, and FCNN). This model was constructed in a novel architecture to investigate the idea that combining different DL techniques can be more effective in predicting cryptocurrencies prices when compared to vanilla DL techniques. The overall average score for MAPE was 9.63, RMSE 4935.29, and 90.37% Accuracy for Bitcoin. However, compared to the

base model, this model underperformed for Ether and with a score for MAPE of 5.72% higher than the based model. The RMSE score was 774.03, again 4.35% higher than the base model. Starting from 35 to 56 days forecasting windows, the combined model one outperformed all other models. CNN, LSTM, and DNNs were cooperative in their modeling capabilities. CNNs reduce frequency variations effectively, LSTMs do well with temporal modelling, and DNNs perform well at mapping features into a more separable domain. This finding supports the results produced by Lu et al. (2020) and Sainath et al. (2015). The results suggest that combining different DL techniques in a carefully designed model while leveraging on the strengths of each technique and at the same time taking care of the known weaknesses of these techniques is indeed a highly effective hybrid architecture to improve the forecasting accuracy. The nearest contender to the combined model one was the LSTM model, especially for larger forecasting horizons.

Combined model Two (CNN, BLSTM and FCNN)

In this model architecture, Bi-directional LSTM replaced the LSTM. Table 4.8 shows that for Bitcoin, this model scored 9.78 for MAPE, 4948.22 for RMSE, and 90.22% for an accurate forecast. Although it outperformed the baseline model for Bitcoin, it underperformed for Ether with results of 16.28 MAPE, 781.66 RMSE, and 83.72% accurate forecasting. This model did not make any significant difference for both currencies.

Comparing to other studies

Finding similar studies to compare the current study results with was quite difficult because the topic is broad and provides infinite options to experiment with. This study evaluated some of the most successful and widely used DL algorithms forecasting cryptocurrency prices in spite of the general misconception that they are inefficient when used for time series forecasting (P. Pintelas et al., 2020). In contradiction to these views, this study results indicate that the evaluated DL techniques produced 85% - 90% for Bitcoin and 80%-85% for Ether accurate forecast, respectively. Although the research methodology had known limitations, such as using the iterative single-step method, skipping the time series analysis stage, and feature engineering, the results obtained provide considerable evidence that DL models are efficient and effective for handling cryptocurrency time series problems. In addition, DL models have proven that they are capable of learning representations out of raw input data.

FCNN was found to be a reasonable and straightforward technique for benchmarking the performance of DL models. This finding supports the suggestions made by Bandara et al. (2020) and Nado et al. (2021). More innovative techniques to improve FCNN architecture to make it a sound baseline model and more empirical comparative studies are still required. It is recommended that for long-term forecasting windows to use the direct, multi-step forecasting approach instead of the iterative single-step approach (Ben Taieb, 2010). However, the results of this study indicate that this could be more practical than a single-step method for expansive forecasting windows but not for short-term forecasting windows, which is the case of unstable cryptocurrencies prices. Therefore, empirical studies are needed to determine if this multi-forecast approach will be more accurate than the iterative one.

An approach developed by Patel et al. (2020) focuses on Litecoin and Monero to predict cryptocurrency prices. The proposed hybrid model outperformed conventional LSTM networks, they concluded. A number of studies have shown that combining DL techniques into hybrid models can improve time series forecasting (Kim & Cho, 2019; Sainath et al., 2015; Li & Dai, 2020). In this regard, the current study tends to agree with that statement because CNN, LSTM, and FCNN (Combine Model 1) provide the most accurate predictions for Bitcoin compared to wider forecasting windows.

5.2 Research Limitations and Weaknesses

The current study focused on the well-known and successful DL techniques such as FCNN, LSTM, and CNN and excluded the latest emerging techniques that are just recently been used for cryptocurrency forecasting problems such as Temporal Convolutional Networks (TCN) and Transformers. This work covered only the time series forecasting process and excluded the time series analysis process. Univariate time series problem was considered, while a multivariate time series problem was excluded. Single-step, recursive forecasting strategy was used instead of a multi-step forecasting strategy. Finally, only the top two cryptocurrencies, Bitcoin and Ether, were considered, while more cryptocurrencies could have been considered to enrich the study.

5.3 Conclusion

Cryptocurrency policymakers, regulators, and investors need accurate prediction tools to support them in making their strategic decisions for assessing cryptocurrency market behavior. Time-series modeling, analysis, and forecasting have historically been critical areas of academic research. In time series analysis and forecasting cryptocurrencies, datasets constitute a challenging task because they are influenced by many uncontrollable factors, volatility, and significant fluctuations. Comparative and empirical studies are becoming essential to better understand the scientific advances in this field to provide a comprehensive and practical comparisons framework for cryptocurrencies forecasting using DL techniques. Identifying the most straightforward, fastest, and best DL technique for each cryptocurrency time series problem is imperative. The current study has argued that deep learning techniques are practical and efficient for predicting cryptocurrencies' prices. With deep learning algorithms

becoming more mature and accurate, forecasting the future is always on the list of uses for them.

The main contribution of the current study to the state of knowledge on this topic was to provide evidence that DL techniques are becoming a preferred tool for different time series problems and, more specifically, for addressing cryptocurrencies' time series forecasting problems due to their advanced capability in learning from raw data. Furthermore, this study suggests that combining DL techniques for cryptocurrency price forecasting will produce the best results for Bitcoin and less optimal results for Ether. In line with the directions and findings of other studies on this topic, the current study provided a comprehensive comparison between different DL categories such as MLP, RNN, CNN, and their combinations.

In conclusion, it was found that the combined, i.e., hybrid, architectures are efficient and provide a more accurate forecast for cryptocurrencies' price when compared to plain vanilla architectures. However, these results cannot be generalized because different cryptocurrency datasets may need carefully designed hybrid models to achieve optimal results. To model a time series forecast, one must consider both the trends and seasonal variations, as well as the correlation between values near each other in DL.

5.4 Practical Recommendations

A variety of machine learning tasks have been successfully addressed with deep learning techniques in recent years, outperforming traditional and machines learning models. In addition to combining powerful DL techniques, it is also essential to develop highly sophisticated and robust

DL baseline architectures that can serve as the benchmark for innovative hybrid techniques in the future. Cryptocurrency time series data preparation is another critical area that requires further exploration. Some researchers recommended skipping the time series analysis and data preparation stages and focus on directly on the time series forecasting stage while capitalizing on the DL skills to learn from raw data. The current study recommends conducting more studies on using DL techniques for time series analysis, data preparation, and forecasting in a single model instead of skipping critical stages in specific time series problems.

In addition to exploring more cryptocurrencies, the methodology used in this study could be easily enhanced to include more cryptocurrency features to deal with multivariate time series problems. It would be beneficial to explore the potential use of cryptocurrency blockchain information and other features such as the daily volume of trades or even economics and technical trading indicators in future research. Additionally, more empirical comparison studies are suggested to uncover the effectiveness of the state-of-the-art DL techniques for time series analysis and forecasting in a different field. Moreover, improving forecast strategies for short and long-term forecast windows could be another interesting direction for future research. Combining iterative single steps with direct multi-step forecasting methods is fundamental to successfully applying DL for time series. For example, DL techniques can be used to find the optimal function that links the forecast horizon, forecast method, and residual error optimization for different cryptocurrencies' time series problems.

References

- Abbasimehr, H., Paki, R., & Bahrini, A. (2021). A novel approach based on combining deep learning models with statistical methods for COVID-19 time series forecasting. *Neural Comput Appl*, 1-15. <https://doi.org/10.1007/s00521-021-06548-9>
- AD, D. (2016). Gold price prediction using support vector regression and ANFIS models. In: . 2016 International conference on computer communication and informatics (ICCCI), IEEE, pp 1–6,
- Aggarwal, A., Gupta, I., Garg, N., & Goel, A. (2019). Deep learning approach to determine the impact of socio economic factors on bitcoin price prediction. 2019 Twelfth International Conference on Contemporary Computing (IC3).
- Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6), 594-621.
- Ai, Y., Li, Z., Gan, M., Zhang, Y., Yu, D., Chen, W., & Ju, Y. (2019). A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system. *Neural Computing and Applications*, 31(5), 1665-1677.
- Altan, A., Karasu, S., & Bekiros, S. (2019). Digital currency forecasting with chaotic meta-heuristic bio-inspired signal processing techniques. *Chaos, Solitons & Fractals*, 126, 325-336.
- Ang, J.-S., Ng, K.-W., & Chua, F.-F. (2020). Modeling Time Series Data with Deep Learning: A Review, Analysis, Evaluation and Future Trend. 2020 8th International Conference on Information Technology and Multimedia (ICIMU).
- Antonopoulos, A. M. (2017). *Mastering Bitcoin: Programming the Open Blockchain*. O'Reilly Media. <https://books.google.com.bh/books?id=tponDwAAQBAJ>
- Antonopoulos, A. M., & D, G. W. P. (2018). *Mastering Ether: Building Smart Contracts and DApps*. O'Reilly Media. <https://books.google.com.bh/books?id=oJJ5DwAAQBAJ>
- Asadi, R., & Regan, A. C. (2020). A spatio-temporal decomposition based deep neural network for time series forecasting. *Applied Soft Computing*, 87, 105963.
- Askari, M., & Askari, H. (2011). Time series grey system prediction-based models: Gold price forecasting. *Trends in Applied Sciences Research*, 6(11), 1287.
- Atsalakis, G. S., Atsalaki, I. G., Pasiouras, F., & Zopounidis, C. (2019). Bitcoin price forecasting with neuro-fuzzy techniques. *European Journal of Operational Research*, 276(2), 770-780.
- Ayoobi, N., Sharifrazi, D., Alizadehsani, R., Shoeibi, A., Gorriz, J. M., Moosaei, H., Khosravi, A., Nahavandi, S., Gholamzadeh Chofreh, A., Goni, F. A., Klemes, J. J., & Mosavi, A. (2021). Time series forecasting of new cases and new deaths rate for COVID-19 using deep learning methods. *Results Phys*, 27, 104495. <https://doi.org/10.1016/j.rinp.2021.104495>
- Bandara, K., Bergmeir, C., & Smyl, S. (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, 140, 112896.
- Baur, D. G., & McDermott, T. K. (2010). Is gold a safe haven? International evidence. *J Bank Finance*, 34. <https://doi.org/10.1016/j.jbankfin.2009.12.008>

- Ben Taieb, S. S., Antti & Bontempi, Gianluca. (2010). Multiple-output modeling for multi-step-ahead time series forecasting. *Neurocomputing.*, 73. 1950-1957.(10.1016/j.neucom.2009.11.030.).
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal Mach Intell*, 35. <https://doi.org/10.1109/TPAMI.2013.50>
- Bontempi, G., Taieb, S. B., & Le Borgne, Y.-A. (2012). Machine learning strategies for time series forecasting. European business intelligence summer school,
- Brownlee, J. (2016). Machine learning mastery with python. *Machine Learning Mastery Pty Ltd*, 527, 100-120.
- Brownlee, J. (2017). *Introduction to Time Series Forecasting With Python: How to Prepare Data and Develop Models to Predict the Future*. Machine Learning Mastery. <https://books.google.com.bh/books?id=-AiqDwAAQBAJ>
- Brownlee, J. (2017). *Long short-term memory networks with python: develop sequence prediction models with deep learning*. Machine Learning Mastery.
- Brownlee, J. (2018). *Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery.
- Chollet, F. (2017). *Deep Learning with Python*. Manning Publications Co.
- Choudhry, S. S., Hassan, T., & Shabi, S. (2015). Relationship between gold and stock markets during the global financial crisis: evidence from nonlinear causality tests. *Int Rev Financ Anal*, 41. <https://doi.org/10.1016/j.irfa.2015.03.011>
- Chowdhury, R., Rahman, M. A., Rahman, M. S., & Mahdy, M. (2020). An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning. *Physica A: Statistical Mechanics and its Applications*, 551, 124569.
- Danial, K. (2019). *Cryptocurrency Investing For Dummies*. Wiley. <https://books.google.com.bh/books?id=rnqGDwAAQBAJ>
- Danial, K., Laurence, T., Kent, P., Bain, T., & Solomon, M. G. (2021). *Cryptocurrency All-in-One For Dummies*. Wiley. <https://books.google.com.bh/books?id=CGJVEAAAQBAJ>
- Daniel, G. (2013). *Principles of artificial neural networks*. World Scientific.
- Demertzis K, I. L., Anezakis VD. (2017). A deep spiking machine-hearing system for the case of invasive fish species. In: 2017 IEEE International conference on innovations in intelligent systems and applications (INISTA), IEEE, pp 23–28.
- Demertzis, K., Iliadis, L., & Bougoudis, I. (2019). Gryphon: a semi-supervised anomaly detection system based on one-class evolving spiking neural network. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-019-04363-x>
- Deng, L., & Platt, J. (2014). Ensemble deep learning for speech recognition. Proc. interspeech,
- Deng, N., Tian, Y., & Zhang, C. (2012). *Support vector machines: optimization based theory, algorithms, and extensions*. Chapman and Hall/CRC. <https://doi.org/10.1201/b14297>
- Derbentsev, V., Babenko, V., Khrustalev, K., Obruch, H., & Khrustalova, S. (2021). Comparative performance of machine learning ensemble algorithms for forecasting cryptocurrency prices. *International Journal of Engineering*, 34(1), 140-148.

- Derbentsev, V., Matviychuk, A., & Soloviev, V. N. (2020). Forecasting of cryptocurrency prices using machine learning. In *Advanced Studies of Financial Technologies and Cryptocurrency Markets* (pp. 211-231). Springer.
- Dias, J. M., Wegrzyn-Wolska, K., Rached, I., González-Vélez, H., Senkerik, R., Pop, C., Cioara, T., Salomie, I., & Bracciali, A. (2019). Forecasting cryptocurrency value by sentiment analysis: An hpc-oriented survey of the state-of-the-art in the cloud era. *High-Performance Modelling and Simulation for Big Data Applications*, 325.
- Dingli, A., & Fournier, K. S. (2017). Financial time series forecasting—a deep learning approach. *International Journal of Machine Learning and Computing*, 7(5), 118-122.
- Ehrlich, S. (2022). *The cryptocurrency Market Crash - A Crypto winter*. <https://www.forbes.com/newsletters/forbescryptoassetadvisor/?sh=d0a95d3dec83>
- Elrom, E. (2019). *The Blockchain Developer: A Practical Guide for Designing, Implementing, Publishing, Testing, and Securing Distributed Blockchain-based Projects*. Apress. <https://books.google.com.bh/books?id=6c2kDwAAQBAJ>
- Faloutsos, C., Gasthaus, J., Januschowski, T., & Wang, Y. (2019). Traditional and contemporary approaches to big time series forecasting. Proceedings of the 2019 International Conference on Management of Data.
- Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019). Deep learning for time series classification: a review. *Data Mining Knowledge Discovery*, 33. <https://doi.org/10.1007/s10618-019-00619-1>
- Ferdiansyah, F., Othman, S. H., Radzi, R. Z. R. M., Stiawan, D., Sazaki, Y., & Ependi, U. (2019). A lstm-method for bitcoin price prediction: A case study yahoo finance stock market. 2019 International Conference on Electrical Engineering and Computer Science (ICECOS),
- Freeman, B. S., Taylor, G., Gharabaghi, B., & The, J. (2018). Forecasting air quality time series using deep learning. *J Air Waste Management Association*, 68(8), 866-886. <https://doi.org/10.1080/10962247.2018.1459956>
- Gamboa, J. C. B. (2017). Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*.
- Garraty, J. A. (1973). The New Deal, National Socialism, and the Great Depression. *The American Historical Review*, 907-944.
- Gasparin, A., Lukovic, S., & Alippi, C. (2019). Deep learning for time series forecasting: The electric load case. *arXiv preprint arXiv:1907.09207*.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- Gridin, I. (2021). *Time Series Forecasting using Deep Learning: Combining PyTorch, RNN, TCN, and Deep Neural Network Models to Provide Production-Ready Prediction Solutions (English Edition)*. BPB Publications. <https://books.google.com.bh/books?id=XrVIEAAAQBAJ>
- Grover, A., Kapoor, A., & Horvitz, E. (2015). A deep hybrid model for weather forecasting. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,
- Guha, B., & Bandyopadhyay, G. (2016). Gold price forecasting using ARIMA model. *J Adv Manag Sci*. <https://doi.org/10.12720/joams.4.2.117-121>

- Gulli, A., & Pal, S. (2017). *Deep learning with Keras*. Packt Publishing Ltd.
- Hitam, N. A., & Ismail, A. R. (2018). Comparative Performance of Machine Learning Algorithms for Cryptocurrency Forecasting.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Comput*, 9. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hua, Y. (2020). Bitcoin price prediction using ARIMA and LSTM. E3S Web of Conferences,
- IE, L. (2020). An advanced active set L-BFGS algorithm for training weight-constrained neural networks. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-019-04689-6>.
- Infante, R. (2019). *Building Ether Dapps: Decentralized applications on the Ether blockchain*. Manning. <https://books.google.com.bh/books?id=0TczEAAAQBAJ>
- Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4), 917-963. <https://doi.org/10.1007/s10618-019-00619-1>
- Ji, S., Kim, J., & Im, H. (2019). A comparative study of bitcoin price prediction using deep learning. *Mathematics*, 7(10), 898.
- Jianwei, E., Ye, J., & Jin, H. (2019). A novel hybrid model on the prediction of time series and its application for the gold price analysis and forecasting. *Phys A*, 527.
- Khedr, A. M., Arif, I., P V, P. R., El-Bannany, M., Alhashmi, S. M., & Sreedharan, M. (2021). Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey. *Intelligent Systems in Accounting, Finance and Management*, 28(1), 3-34.
- Kilimci, H., Yıldırım, M., & Kilimci, Z. H. (2021). The Prediction of Short-Term Bitcoin Dollar Rate (BTC/USDT) using Deep and Hybrid Deep Learning Techniques. 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT),
- Kilimci, Z. H. (2020). Sentiment Analysis Based Direction Prediction in Bitcoin using Deep Learning Algorithms and Word Embedding Models. *International Journal of Intelligent Systems and Applications in Engineering*, 8(2), 60-65.
- Kim, T.-Y., & Cho, S.-B. (2019). Predicting residential energy consumption using CNN-LSTM neural networks. *Energy*, 182, 72-81.
- Klaas, J. (2019). *Machine learning for finance: principles and practice for financial insiders*. Packt Publishing Ltd.
- Koprinska, I., Wu, D., & Wang, Z. (2018). Convolutional neural networks for energy time series forecasting. 2018 international joint conference on neural networks (IJCNN),
- Korstanje, J. Advanced Forecasting with Python.
- Korstanje, J. (2021). *Advanced Forecasting with Python: With State-of-the-Art-Models Including LSTMs, Facebook's Prophet, and Amazon's DeepAR*. Apress. <https://books.google.com.bh/books?id=s4xqzgEACAAJ>
- Krizhevsky A, S. I., Hinton G. (2012). ImageNet: classification with deep convolutional neural networks. In: Advances in neural information processing systems, IEEE, pp 1097–1105.
- Krollner, B., Vanstone, B. J., & Finnie, G. R. (2010). Financial time series forecasting with machine learning techniques: a survey. ESANN,

- Kulkarni, K. (2018). *Learn Bitcoin and Blockchain: Understanding blockchain and Bitcoin architecture to build decentralized applications*. Packt Publishing.
<https://books.google.com.bh/books?id=UPFsDwAAQBAJ>
- Kumar, D., & Rath, S. (2020). Predicting the trends of price for Ether using deep learning techniques. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems* (pp. 103-114). Springer.
- Lahmiri, S., & Bekiros, S. (2019). Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons & Fractals*, 118, 35-40.
- Lara-Benitez, P., Carranza-Garcia, M., & Riquelme, J. C. (2021). An Experimental Review on Deep Learning Architectures for Time Series Forecasting. *Int J Neural Syst*, 31(3), 2130001.
<https://doi.org/10.1142/S0129065721300011>
- Lara-Benítez, P., Carranza-García, M., & Riquelme, J. C. (2021). An Experimental Review on Deep Learning Architectures for Time Series Forecasting. *arXiv preprint arXiv:2103.12057*.
- Lazzeri, F. (2020). *Machine Learning for Time Series Forecasting with Python*. John Wiley & Sons.
- Li, J., Dai, Q., & Ye, R. (2018). A novel double incremental learning algorithm for time series prediction. *Neural Comput Appl*, 31. <https://doi.org/10.1007/s00521-018-3434-0>
- Li, T., Hua, M., & Wu, X. (2020). A hybrid CNN-LSTM model for forecasting particulate matter (PM_{2.5}). *IEEE Access*, 8, 26933-26940.
- Li, Y., & Dai, W. (2020). Bitcoin price forecasting method based on CNN-LSTM hybrid neural network model. *The Journal of Engineering*, 2020(13), 344-347.
- Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philos Trans A Math Phys Eng Sci*, 379(2194), 20200209. <https://doi.org/10.1098/rsta.2020.0209>
- Lin, T., Guo, T., & Aberer, K. (2017). Hybrid neural networks for learning the trend in time series. *Proceedings of the twenty-sixth international joint conference on artificial intelligence*,
- Liping X, M. L. (2011). Short-term analysis and prediction of gold price based on ARIMA model. *Finance Econ* 1.
- Liu D, L. Z. (2017). Gold price forecasting and related influence factors analysis based on random forest. In: *Proceedings of the 10th international conference on management science and engineering management*, Springer, pp 711–723.
- Livieris, I. E., Kiriakidou, N., Stavroyiannis, S., & Pintelas, P. (2021). An advanced CNN-LSTM model for cryptocurrency forecasting. *Electronics*, 10(3), 287.
- Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. *Neural Computing and Applications*, 32(23), 17351-17360.
<https://doi.org/10.1007/s00521-020-04867-x>
- Livieris, I. E., Pintelas, E., Stavroyiannis, S., & Pintelas, P. (2020). Ensemble deep learning models for forecasting cryptocurrency time-series. *Algorithms*, 13(5), 121.
- Livieris, I. E., Stavroyiannis, S., Iliadis, L., & Pintelas, P. (2021). Smoothing and stationarity enforcement framework for deep learning time-series forecasting. *Neural Comput Appl*, 1-15.
<https://doi.org/10.1007/s00521-021-06043-1>

- Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. *Complexity*, 2020.
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLoS One*, 13(3), e0194889.
- Makridou, G., Atsalakis, G. S., Zopounidis, C., & Andriosopoulos, K. (2013). Gold price forecasting with a neuro-fuzzy-based inference. *Int J Financ Eng Risk Manag*, 1. <https://doi.org/10.1504/IJFERM.2013.053707>
- Mather, B. (2020). *Time Series with Python: How to Implement Time Series Analysis and Forecasting Using Python*. Abiproduct Pty Limited. <https://books.google.com.bh/books?id=Cah8zQEACAAJ>
- McNally, S., Roche, J., & Caton, S. (2018). Predicting the price of bitcoin using machine learning. 2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP),
- Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2011). *Introduction to Time Series Analysis and Forecasting*. Wiley. <https://books.google.com.bh/books?id=-qaFi0oOPAYC>
- Mougayar, W., & Buterin, V. (2016). *The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology*. Wiley. <https://books.google.com.bh/books?id=CEsPDAAAQBAJ>
- Nado, Z., Band, N., Collier, M., Djolonga, J., Dusenberry, M. W., Farquhar, S., Filos, A., Havasi, M., Jenatton, R., & Jerfel, G. (2021). Uncertainty Baselines: Benchmarks for Uncertainty & Robustness in Deep Learning. *arXiv preprint arXiv:2106.04015*.
- Nielsen, A. (2019). *Practical Time Series Analysis: Prediction with Statistics and Machine Learning*. O'Reilly Media. <https://books.google.com.bh/books?id=xNOwDwAAQBAJ>
- Nosratabadi, S., Mosavi, A., Duan, P., Ghamisi, P., Filip, F., Band, S. S., Reuter, U., Gama, J., & Gandomi, A. H. (2020). Data science in economics: comprehensive review of advanced machine learning and deep learning methods. *Mathematics*, 8(10), 1799.
- Pal, A., & Prakash, P. (2017). *Practical Time Series Analysis: Master Time Series Data Processing, Visualization, and Modeling using Python*. Packt Publishing. <https://books.google.com.bh/books?id=EJpGDwAAQBAJ>
- Patel, M. M., Tanwar, S., Gupta, R., & Kumar, N. (2020). A deep learning-based cryptocurrency price prediction scheme for financial institutions. *Journal of Information Security and Applications*, 55, 102583.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., & Dubourg, V. (2011). Scikit-learn: machine learning in python. *J Mach Learn Res*, 12.
- Pintelas, E., Livieris, I. E., Stavroyiannis, S., Kotsilieris, T., & Pintelas, P. (2020). Investigating the problem of cryptocurrency price prediction: a deep learning approach. IFIP International Conference on Artificial Intelligence Applications and Innovations,
- Pintelas, P., Kotsilieris, T., Livieris, I., Pintelas, E., & Stavroyiannis, S. (2020). *Fundamental research questions and proposals on predicting cryptocurrency prices using DNNs*.

- Politis, A., Doka, K., & Koziris, N. (2021). Ether price prediction using advanced deep learning models. 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC),
- Qin, D., Yu, J., Zou, G., Yong, R., Zhao, Q., & Zhang, B. (2019). A novel combined prediction scheme based on CNN and LSTM for urban PM 2.5 concentration. *IEEE Access*, 7, 20050-20059.
- Qiu, X., Ren, Y., Suganthan, P. N., & Amaratunga, G. A. (2017). Empirical mode decomposition based ensemble deep learning for load demand time series forecasting. *Applied Soft Computing*, 54, 246-255.
- Qiu, X., Zhang, L., Ren, Y., Suganthan, P. N., & Amaratunga, G. (2014). Ensemble deep learning for regression and time series forecasting. 2014 IEEE symposium on computational intelligence in ensemble learning (CIEL),
- Rahmani Cherati, M., Haeri, A., & Ghannadpour, S. F. (2021). Cryptocurrency direction forecasting using deep learning algorithms. *Journal of Statistical Computation and Simulation*, 1-15.
- Rangapuram, S. S., Seeger, M. W., Gasthaus, J., Stella, L., Wang, Y., & Januschowski, T. (2018). Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31, 7785-7794.
- Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: a comprehensive review. *Neural Comput*, 29. https://doi.org/10.1162/neco_a_00990
- Reid, D., Jaafar, H. A., & Hissam, T. (2014). Financial time series prediction using spiking neural networks. *PloS One*, 9. <https://doi.org/10.1371/journal.pone.0103656>
- Roy, S., Nanjiba, S., & Chakrabarty, A. (2018). Bitcoin price forecasting using time series analysis. 2018 21st International Conference of Computer and Information Technology (ICCIT),
- Sadaei, H. J., e Silva, P. C. d. L., Guimarães, F. G., & Lee, M. H. (2019). Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy*, 175, 365-377.
- Sainath, T. N., Vinyals, O., Senior, A., & Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks. 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP),
- Salis VE, K. A., Singh A. (2019). Prediction of gold stock market using hybrid approach. In: Emerging research in electronics, computer science and technology, Springer, pp 803–812.
- Schliebs, S., & Kasabov, N. (2013). Evolving spiking neural network: a survey. *Evol Syst*, 4. <https://doi.org/10.1007/s12530-013-9074-9>
- Sebastião, H., & Godinho, P. (2021). Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financial Innovation*, 7(1), 1-30.
- Seo, Y., & Hwang, C. (2018). Predicting bitcoin market trend with deep learning models. *Quantitative Bio-Science*, 37(1), 65-71.
- Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90, 106181.
- Shafiee, S., & Topal, E. (2010). An overview of global gold market and gold price forecasting. *Resour Policy*, 35. <https://doi.org/10.1016/j.resourpol.2010.05.004>

- Shastri, S., Singh, K., Kumar, S., Kour, P., & Mansotra, V. (2020). Time series forecasting of Covid-19 using deep learning models: India-USA comparative case study. *Chaos Solitons Fractals*, 140, 110227. <https://doi.org/10.1016/j.chaos.2020.110227>
- Shen, Z., Zhang, Y., Lu, J., Xu, J., & Xiao, G. (2020). A novel time series forecasting model with deep learning. *Neurocomputing*, 396, 302-313.
- Shintate, T., & Pichl, L. (2019). Trend prediction classification for high frequency bitcoin time series with deep learning. *Journal of risk and Financial management*, 12(1), 17.
- Singh, A., Kumar, A., & Akhtar, Z. (2021). Bitcoin Price Prediction: A Deep Learning Approach. 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN),
- Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1), 75-85.
- Spadon, G., Hong, S., Brandoli, B., Matwin, S., Rodrigues-Jr, J. F., & Sun, J. (2021). Pay Attention to Evolution: Time Series Forecasting with Deep Graph-Evolution Learning. *IEEE Trans Pattern Anal Mach Intell*, PP. <https://doi.org/10.1109/TPAMI.2021.3076155>
- Strickland, J. (2020). *Time Series Analysis and Forecasting Using Python* & R. Lulu.com. <https://books.google.com.bh/books?id=8WknzgEACAAJ>
- Swing, C., & Nakamoto, M. (2021). *Bitcoin and Cryptocurrency 2021 (2 Books in 1): The Ultimate Crypto Trading and Investing Guide to Profit During the 2021 Bull Run and Change Your Life!* Online Creative Services Di Daniel Zanatta. <https://books.google.com.bh/books?id=KetlzgEACAAJ>
- Taieb, S. B., & Hyndman, R. J. (2012). *Recursive and direct multi-step forecasting: the best of both worlds* (Vol. 19). Citeseer.
- Tapscott, D., & Tapscott, A. (2018). *Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies Is Changing the World*. Penguin Publishing Group. <https://books.google.com.bh/books?id=8qlPEAAAQBAJ>
- Thanasis, Z., Elias, H., & Manolis, V. (2020). ETH analysis and predictions utilizing deep learning. *Expert Systems with Applications*, 162, 113866. <https://doi.org/https://doi.org/10.1016/j.eswa.2020.113866>
- Tikhomirov, S. (2017). Ether: state of knowledge and research perspectives. International Symposium on Foundations and Practice of Security,
- Torres, J. F., Hadjout, D., Sebaa, A., Martinez-Alvarez, F., & Troncoso, A. (2021). Deep Learning for Time Series Forecasting: A Survey. *Big Data*, 9(1), 3-21. <https://doi.org/10.1089/big.2020.0159>
- Tsihrintzis, G. A., Virvou, M., & Jain, L. C. Advances in Machine Learning/Deep Learning-based Technologies.
- Ur Sami, I. (2017). Predicting future gold rates using machine learning approach. *Int J Adv Comput Sci Appl*, 8.
- Uras, N., & Ortu, M. (2021). Investigation of Blockchain Cryptocurrencies' Price Movements Through Deep Learning: A Comparative Analysis. 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER),

- Vargas, M. R., De Lima, B. S., & Evsukoff, A. G. (2017). Deep learning for stock market prediction from financial news articles. 2017 IEEE international conference on computational intelligence and virtual environments for measurement systems and applications (CIVEMSA),
- Vijayalakshmi, B., Kaliappan, M., Ramar, K., Vijayalakshmi, K., & Vimal, S. (2021). Forecasting Using Deep Learning Approaches. In *Concepts and Real-Time Applications of Deep Learning* (pp. 167-186). Springer.
- Vishwas, B. V., & PATEL, A. (2020). *Hands-on Time Series Analysis with Python: From Basics to Bleeding Edge Techniques*. Apress. <https://books.google.com.bh/books?id=cTeHzQEACAAJ>
- Wang, G. J., Xie, C., Jiang, Z. Q., & Stanley, H. E. (2016). Extreme risk spillover effects in world gold markets and the global financial crisis. *Int Rev Econ Finance*, 46. <https://doi.org/10.1016/j.iref.2016.08.004>
- Wen, F., Yang, X., Gong, X., & Lai, K. K. (2017). Multi-scale volatility feature analysis and prediction of gold price. *Int J Inf Technol Decis Mak*, 16. <https://doi.org/10.1142/S0219622016500504>
- Widiputra, H. M., Adele & Gautama, Elliana. (2021). Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction. *Complexity*. 2021. 10.1155/2021/9903518.
- Wu, C.-H., Lu, C.-C., Ma, Y.-F., & Lu, R.-S. (2018). A new forecasting framework for bitcoin price with LSTM. 2018 IEEE International Conference on Data Mining Workshops (ICDMW),
- Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*,
- Yahoo Finance, (2021) [Online]. Available: <https://finance.yahoo.com/quote/>.
- Yogeshwaran, S., Kaur, M. J., & Maheshwari, P. (2019). Project based learning: Predicting bitcoin prices using deep learning. 2019 IEEE Global Engineering Education Conference (EDUCON),
- Zaini, N., Ean, L. W., Ahmed, A. N., & Malek, M. A. (2021). A systematic literature review of deep learning neural network for time series air quality forecasting. *Environ Sci Pollut Res Int*. <https://doi.org/10.1007/s11356-021-17442-1>
- Zeroual, A., Harrou, F., Dairi, A., & Sun, Y. (2020). Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. *Chaos Solitons Fractals*, 140, 110121. <https://doi.org/10.1016/j.chaos.2020.110121>
- Zhang, H., & Fu, R. (2020). A Hybrid Approach for Turning Intention Prediction Based on Time Series Forecasting and Deep Learning. *Sensors (Basel)*, 20(17). <https://doi.org/10.3390/s20174887>
- Zhang, Z., Dai, H.-N., Zhou, J., Mondal, S. K., García, M. M., & Wang, H. (2021). Forecasting Cryptocurrency Price Using Convolutional Neural Networks with Weighted and Attentive Memory Channels. *Expert Systems with Applications*, 115378.
- Zheng, J., Fu, X., & Zhang, G. (2019). Research on exchange rate forecasting based on deep belief network. *Neural Comput Appl*, 31. <https://doi.org/10.1007/s00521-017-3039-z>
- Zou, W., & Xia, Y. (2019). Back propagation bidirectional extreme learning machine for traffic flow time series prediction. *Neural Comput Appl*, 31. <https://doi.org/10.1007/s00521-018-3578-y>
- Zoumpikas, T., Houstis, E., & Vavalis, M. (2020). ETH analysis and predictions utilizing deep learning. *Expert Systems with Applications*, 162, 113866.

Appendix A: Tools and Libraries

Python

There is tremendous potential for Python to be the most dominant platform for applied machine learning. Python is a flexible general-purpose programming language that can be used for data science projects due to its versatility as a programming language.

Scikit-learn

For practicing and developing machine learning using Python, the scikit-learn library was used. The library is focused on machine learning algorithms for classification, regression, clustering, etc. As well as providing tools for evaluating models, tuning parameters, and preprocessing data, it also features a broad set of related tasks

SciPy

The SciPy library for Python offers an extensive collection of math, science, and engineering tools. For data science projects that are based on Python, there are two SciPy libraries essential. In addition to NumPy, which offers efficient array operations, Matplotlib provides data visualization options.

Pandas

Pandas is a Python library that contains high-performance tools that load and handle data efficiently. For representing data, it provides convenient data structures like DataFrame and Series, which are built upon and require the SciPy ecosystem under the hood. With Pandas library, time series data is especially well supported.

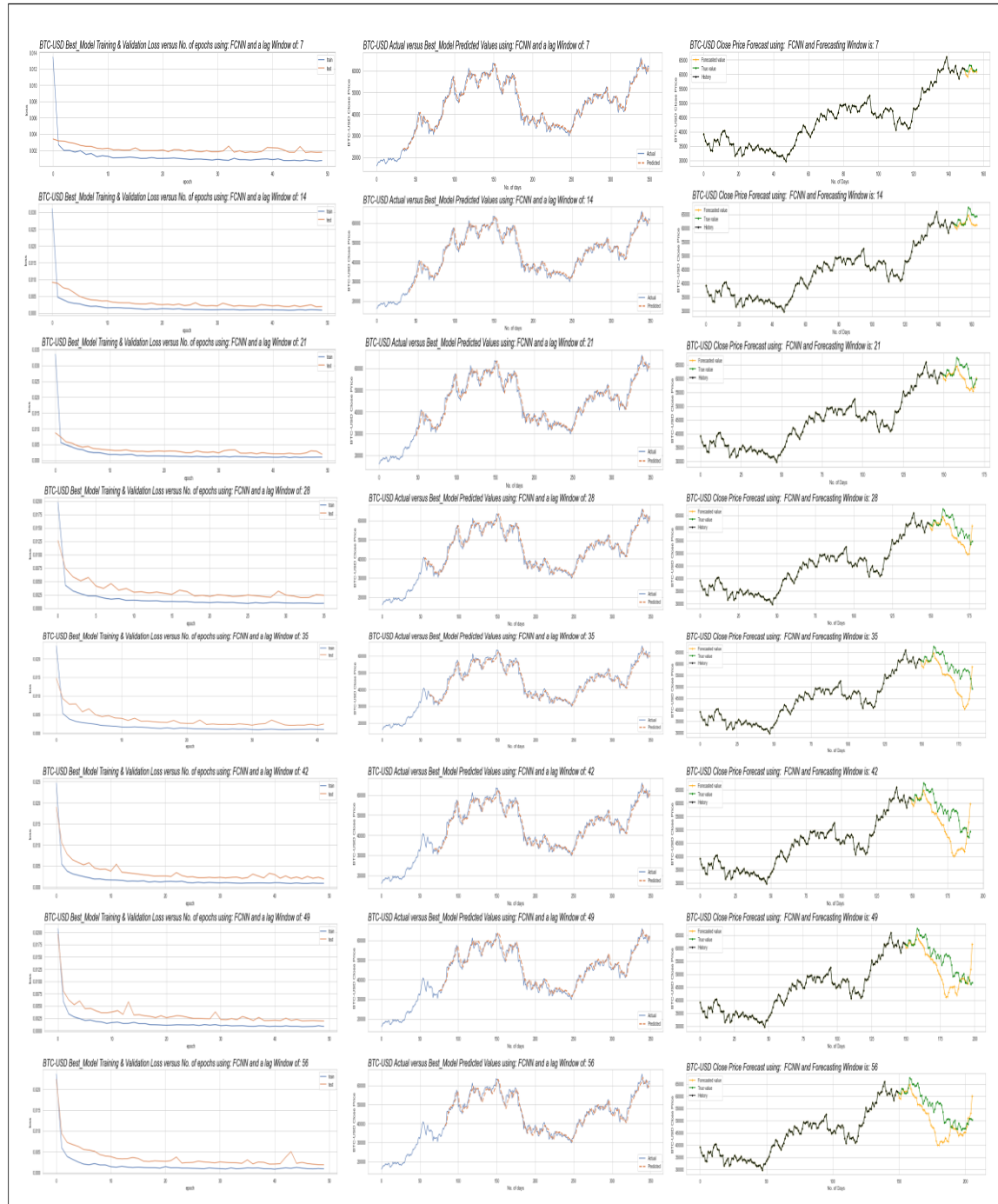
TensorFlow and Keras

The TensorFlow platform is an open-source machine learning platform that provides end-to-end support. Open-source tools and libraries are integrated with this repository. It provides advanced tools for researchers to create and deploy deep learning-powered applications. On the other hand, the Keras framework allows for fast experimenting and integrates seamlessly with TensorFlow. An important benefit is its ability to transform ideas into results quickly.

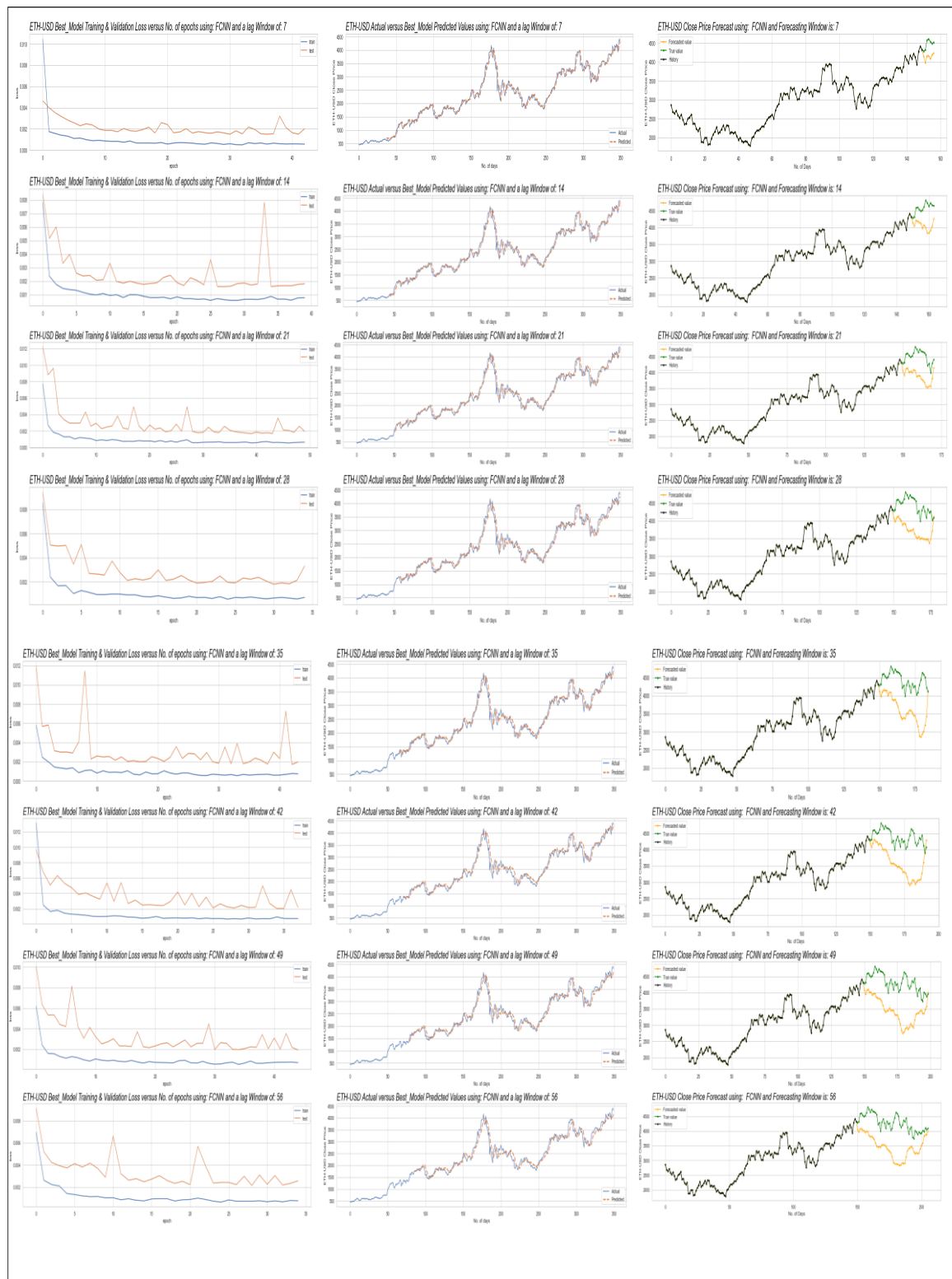
Appendix B: Experiments' Results

The following plots represent the results obtained from the implemented models.

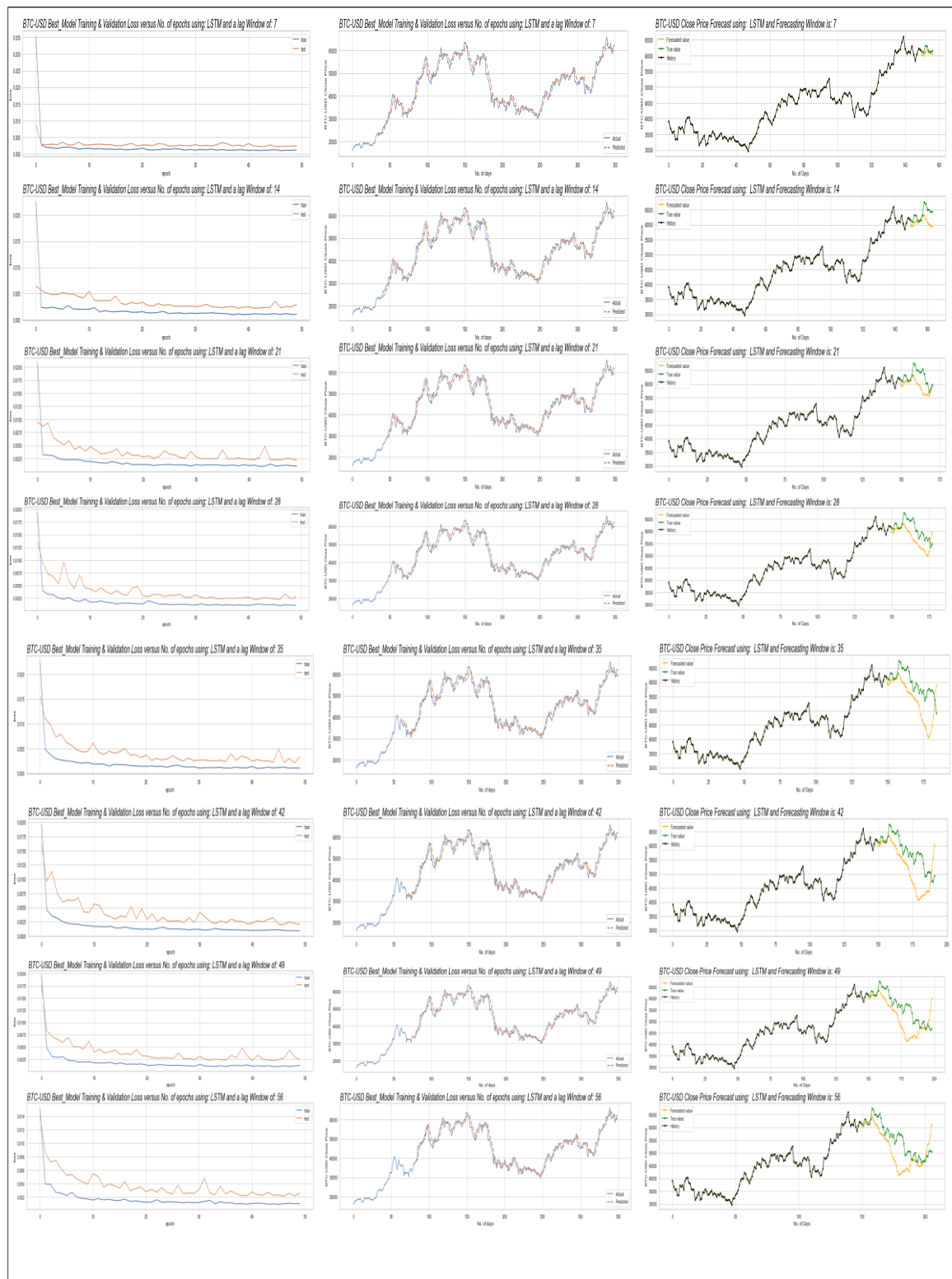
FCNN Results for Bitcoin



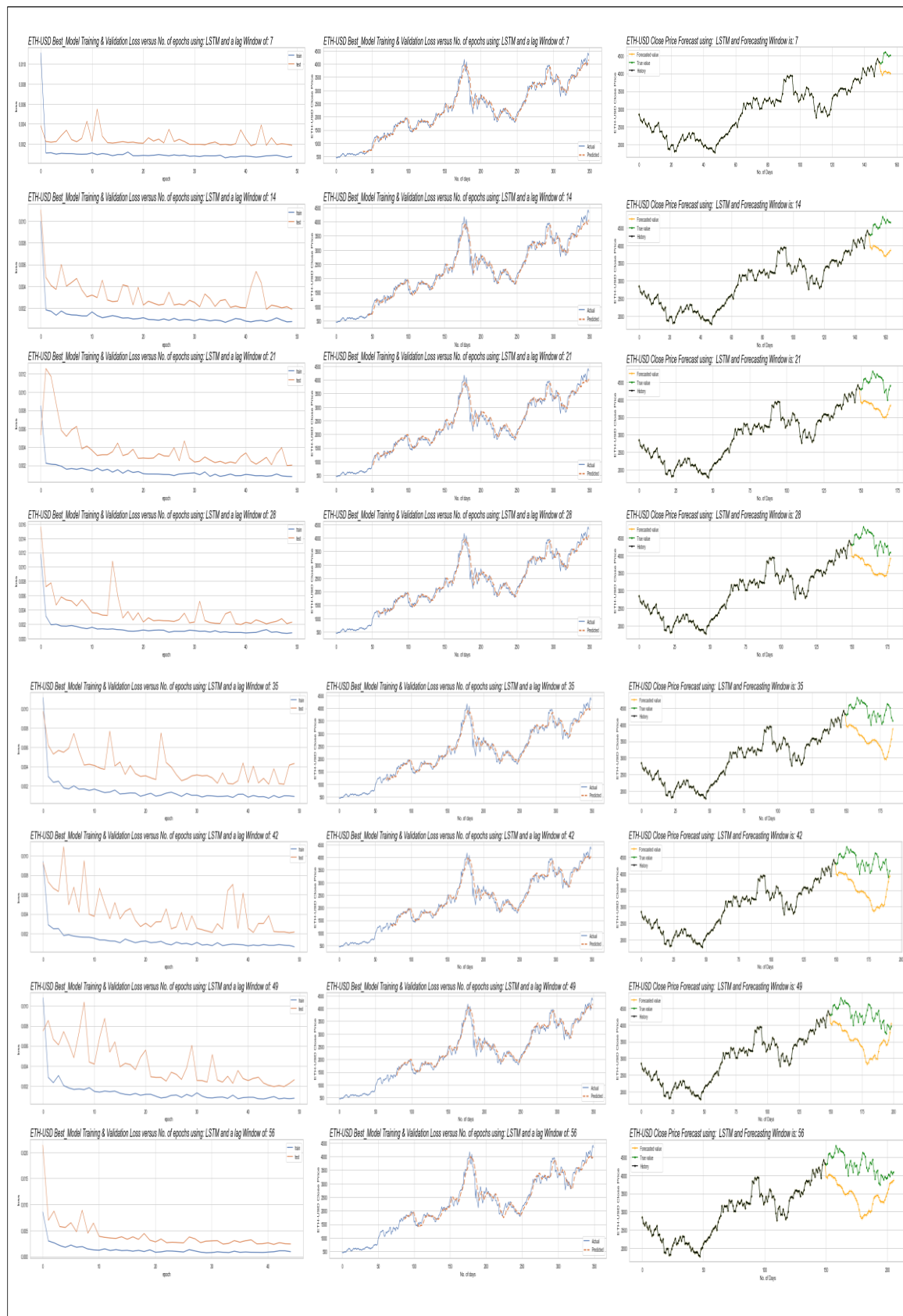
FCNN Results for Ether



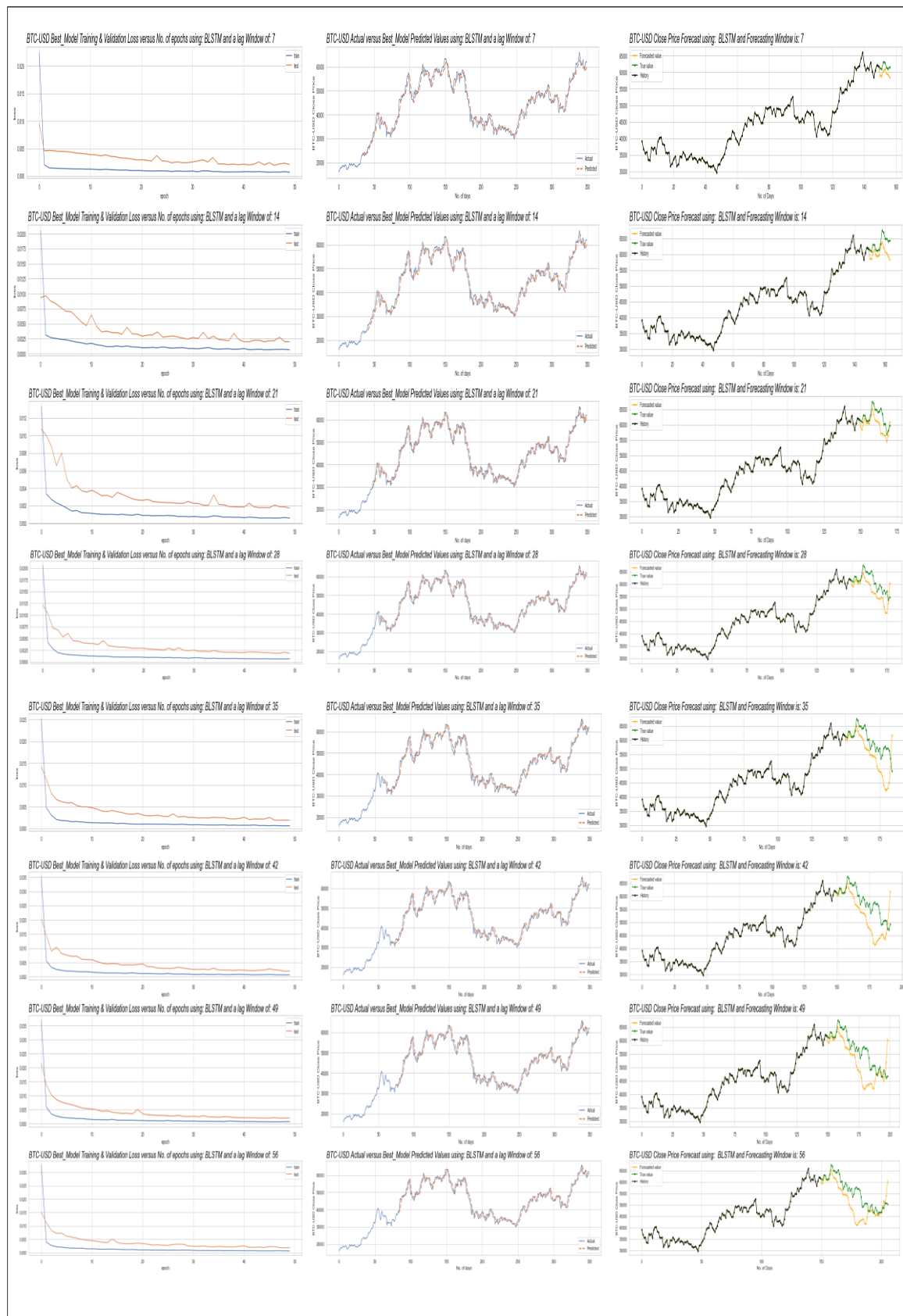
LSTM Results for Bitcoin



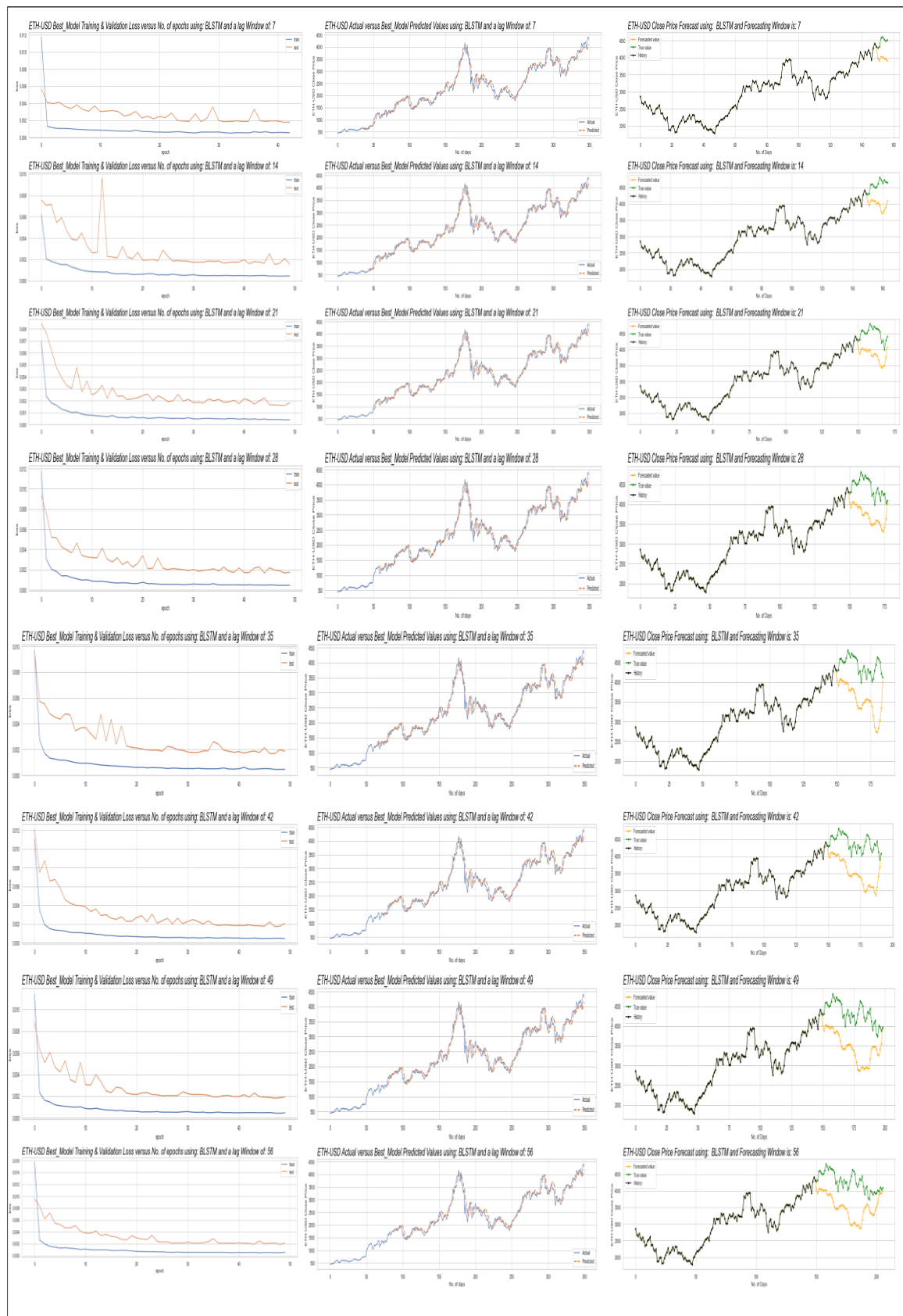
LSTM Results for Ether



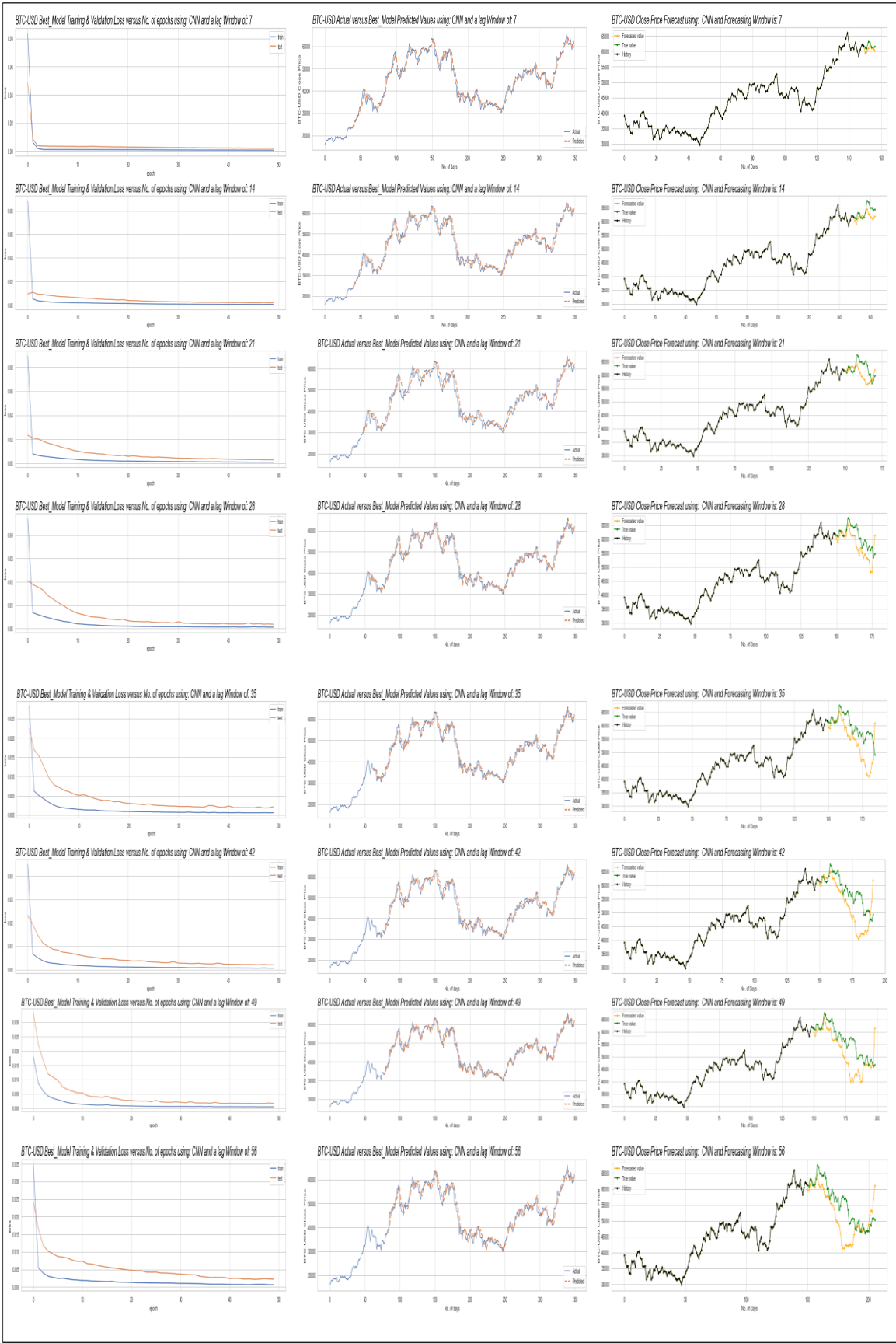
BLSTM Results on Bitcoin



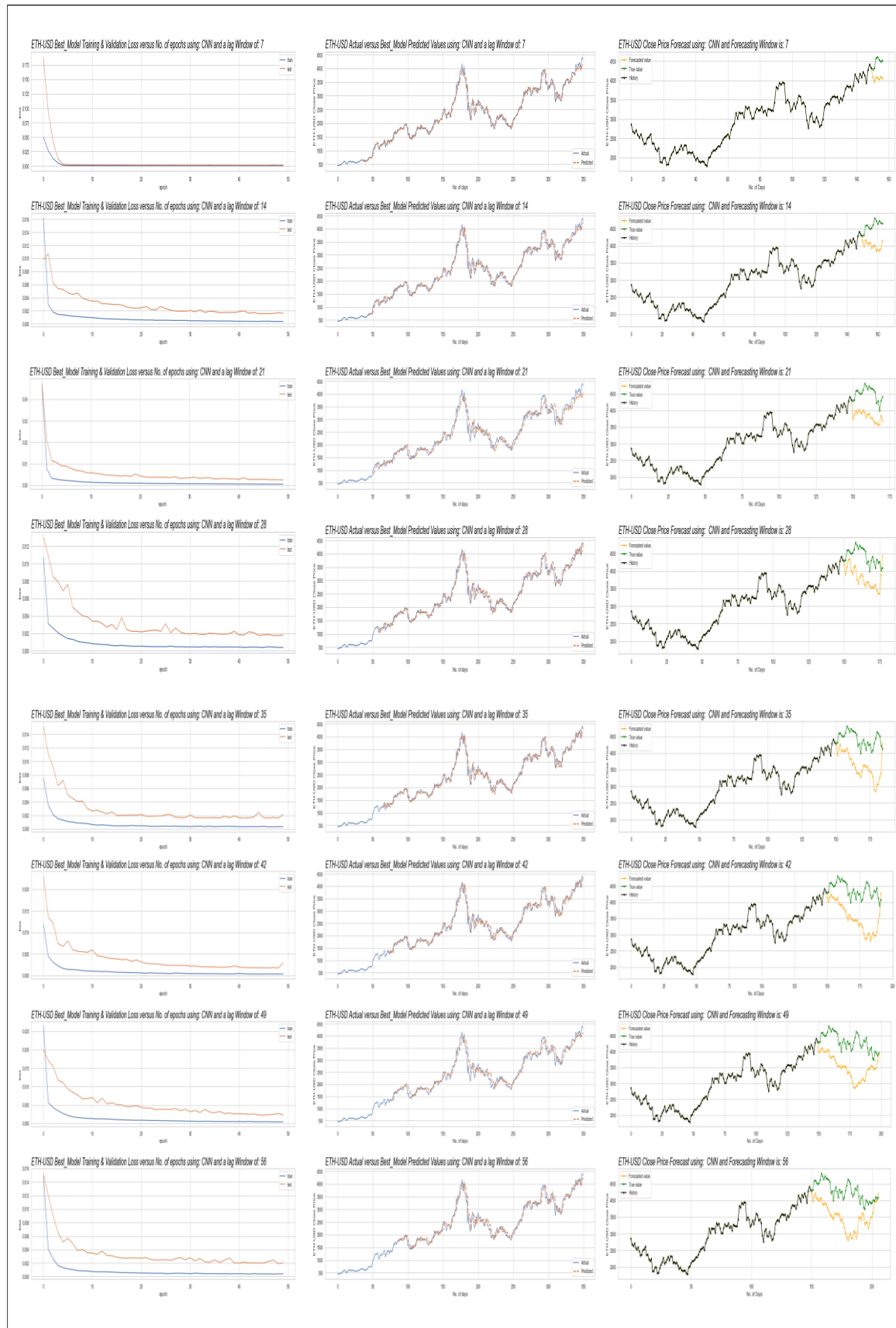
BLSTM Results on Ether



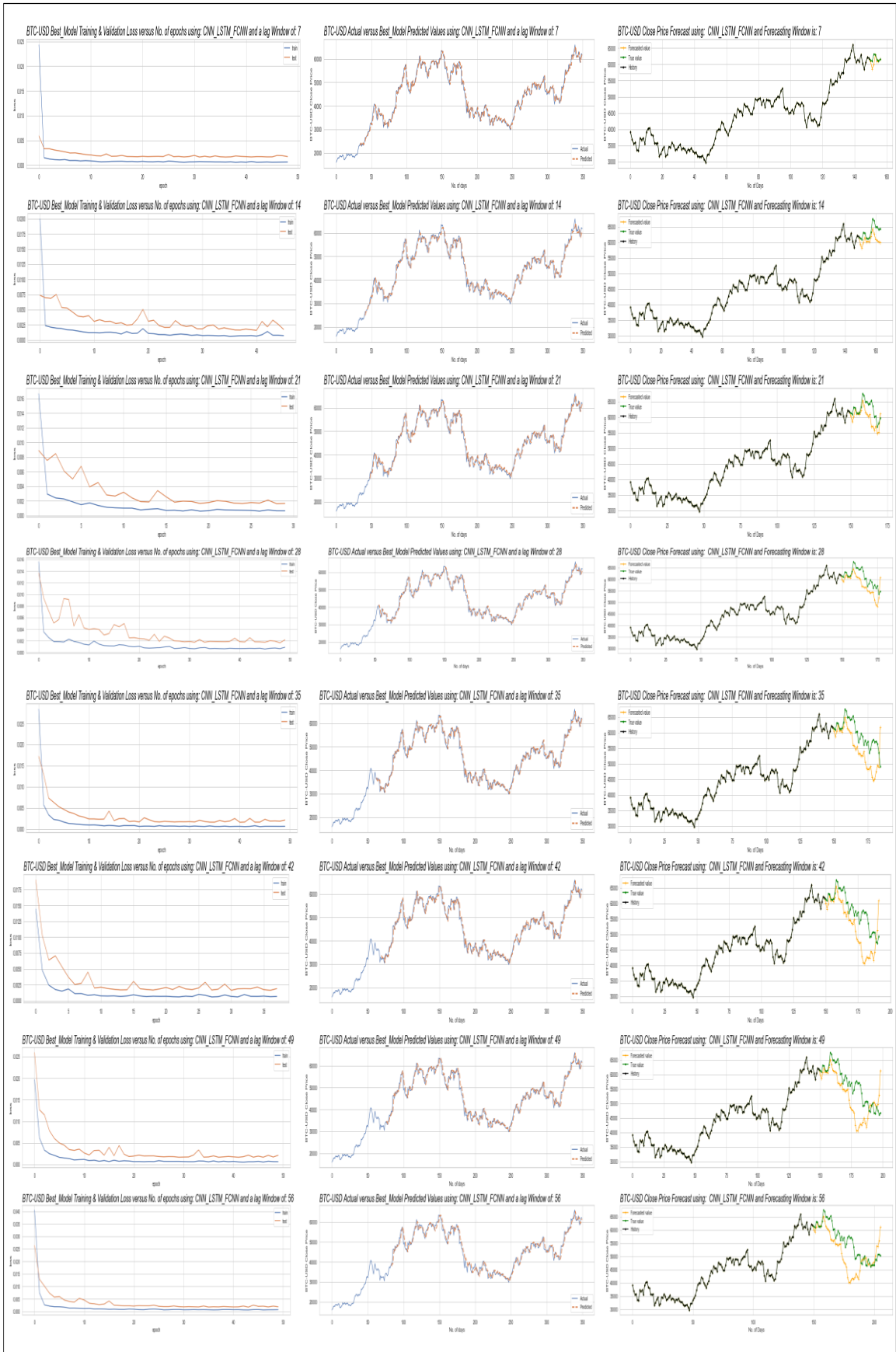
CNN Results on Bitcoin



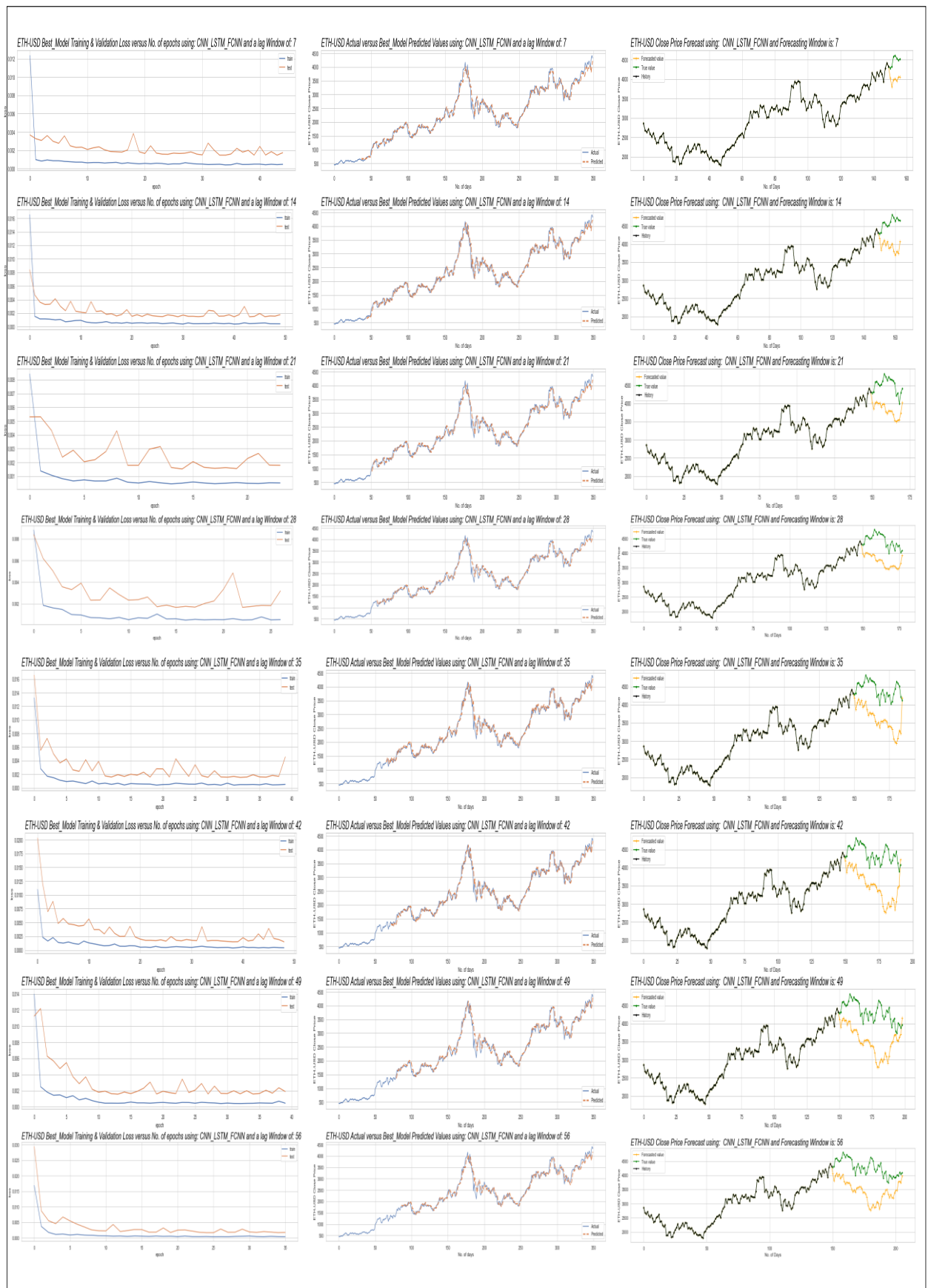
CNN Results on Ether



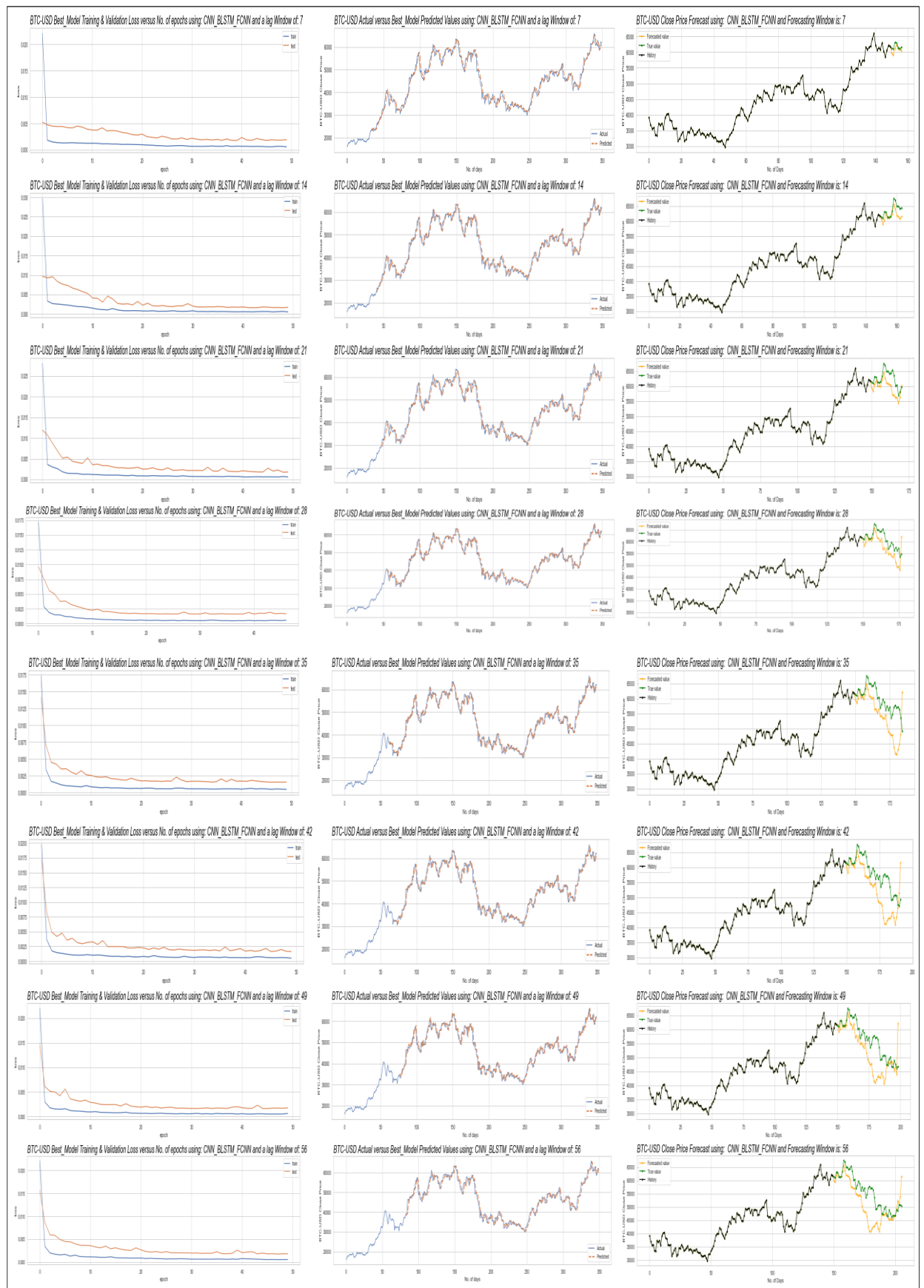
Combined Model One: CNN_LSTM_FCNN Results on Bitcoin



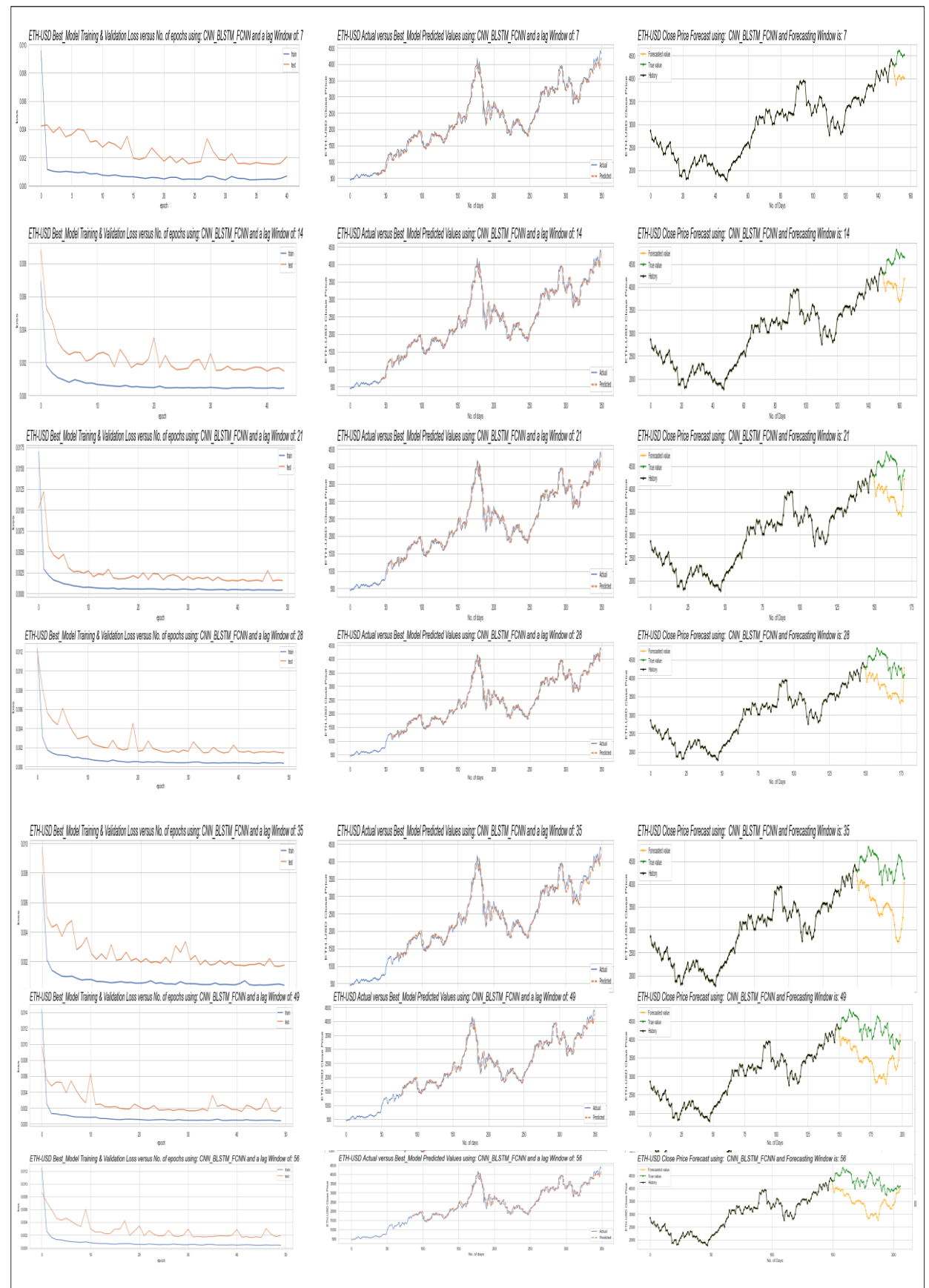
Combined Model One: CNN_LSTM_FCNN Results on Ether



Combined Model Two: CNN_BLSTM_FCNN Results on Bitcoin



Combined Model Two: CNN_BLSTM_FCNN Results on Ether



الملخص

في الآونة الأخيرة، تم تطبيق التعلم العميق وتقديمه في الأدبيات الأكاديمية كبديل محتمل للتعلم الآلي والأساليب الإحصائية التقليدية للتنبؤ بالسلاسل الزمنية. ومع ذلك، هناك حاجة لمزيد من الدراسات المقارنة التجريبية حول فعالية تقنيات التعلم العميق لتحديات التنبؤ المختلفة. يمكن معالجة مشاكل السلاسل الزمنية مثل التبعيات المعقدة غير الخطية والمدخلات متعددة المتغيرات والتنبؤ متعدد الخطوات بكفاءة بواسطة تقنيات التعلم العميق نظرًا لقدرتها القوية على التعلم من البيانات الأولية. تقارن هذه الدراسة التجريبية مدى ملاءمة وأداء 6 تقنيات التعلم العميق. هذه هي: الشبكة العصبية المتصلة بالكامل (FCN)، والذاكرة طويلة المدى (LSTM)، والذاكرة ثنائية الاتجاه طويلة المدى (BLSTM)، والشبكة العصبية التلافيفية (CNN)، واثنين من البنى الهجينة المركبة من هذه التقنيات باستخدام أعلى عملتين مشفرتين حاليين هما Bitcoin و Ether. تشير النتائج إلى أن تقنيات التعلم العميق فعالة في التنبؤ بسعر العملة المشفرة بمتوسط إجمالي MAPE يبلغ 9.76، ومتوسط RMSE يبلغ 5007.75 ومتوسط دقة 90.24٪. علاوة على ذلك، فإن الجمع بين تقنيات DL المختلفة في بنية محسنة يمكن أن يكون أكثر فاعلية للتنبؤ بسعر العملات المشفرة من البنى البسيطة. تشير النتائج إلى أن النموذج المركب رقم 1 (CNN_LSTM_FCNN) بالنسبة إلى Bitcoin تفوق في الأداء على النموذج الأساسي بالإضافة إلى جميع النماذج الأخرى. عند مقارنتها بالنموذج الأساسي FCNN، سجلت نسبة MAPE 2.83٪ و RMSE 3.03٪ أقل. ومع ذلك، بالنسبة إلى Ether، تفق CNN مع أداء النموذج الأساسي بنسبة 0.47٪ أقل من MAPE و 0.54٪ أقل RMSE مقارنة بالنموذج الأساسي. علاوة على ذلك، يمكن أن تدرس الأبحاث المستقبلية كيفية تحسين فعالية استراتيجية التنبؤ وكيفية استخدام DL لتقدير التنبؤ الأمثل للعملات المشفرة المختلفة.

جامعة البحرين

قسم برامج الدراسات العليا

كلية العلوم



التنبأ بأسعار العملات المشفرة باستخدام أنظمة شبكات التعلم العميق – دراسة مقارنة

أطروحة مقدمة كجزء من متطلبات الحصول على درجة
الماجستير
في علوم وتحليل البيانات الضخمة

إعداد

محمد حمد خالد بوبشيت

الرقم الجامعي: 202012424

إشراف

د. نبيل حويحي
أستاذ

د. سوسن هلال
أستاذ مساعد

جامعة البحرين

جامعة البحرين

مملكة البحرين
مارس 2022