METAVERSE-BASED CRYPTOCURRENCIES PREDICTION USING MACHINE LEARNING

LO GUAN SIANG

FACULTY OF COMPUTING AND INFORMATICS

UNIVERSITI MALAYSIA SABAH

2023

LO GUAN SIANG

THESIS SUBMITTED IN PARTIAL FULFILLMENT FOR THE DEGREE OF BACHELOR OF COMPUTER SCIENCE WITH HONOURS (DATA SCIENCE)

FACULTY OF COMPUTING AND INFORMATICS
UNIVERSITI MALAYSIA SABAH

NAME **:** LO GUAN SIANG

MATRIC NUMBER **:** BI19110220

TITLE **:** METAVERSE-BASED CRYPTOCURRENCIES PREDICTION USING MACHINE LEARNING

DEGREE **:** BACHELOR OF COMPUTER WITH HONOURS (DATA SCIENCE)

VIVA'S DATE **:** 26/1/2023

CERTIFIED BY;

1. **SUPPORTED**

1. **SUPERVISOR**	Signature
PROF. DR. JASON TEO TZE WI	

DECLARATION

I hereby declare that the material in this thesis is my own except for quotations, equations, summaries, and references, which have been duly acknowledged.

26 JAN 2023

LO GUAN SIANG
BI19110220

ACKNOWLEDGEMENT

I would like to acknowledge and express my greatest thanks to my supervisor Prof. Dr. Jason Teo Tze Wi. His guidance and advice carried me through all the stages of writing my proposal. Without his guidance and advice, this work would not be possible.

LO GUAN SIANG

26 JAN 2023

ABSTRACT

Cryptocurrencies have recently attracted much interest from investors and researchers. Cryptocurrencies have become a global phenomenon in financial sectors and investors' preferred choice for traded financial instruments due to their simplicity, innovation, security, and transparency, as well as their decentralised nature. Metaverse-based cryptocurrencies are one subtopic use case for cryptocurrencies used for the trading and investing in the digital asset and lands in the metaverse, a virtual world that is an extension of the real world. These cryptocurrencies, including metaverse-based ones, have drastic fluctuations and aggressive movements in their prices, which is highly unpredictable. Although there are some recent studies for machine learning implementation to predict cryptocurrencies, it has never been implemented in Metaverse-based cryptocurrencies. This study will design machine learning models for the price predictions of metaverse-based cryptocurrencies. Furthermore, numerous machine learning algorithms will be implemented as the predictive models for metaverse-based cryptocurrencies' closing prices. The development of the machine learning model will follow the process below: data collection, data pre-processing, model choosing, model training and development and modal evaluation. Lastly, the contribution of this study can conclude as follows. This study aims to help the investor and researcher to minimise the risk in the cryptocurrency market and diversify cryptocurrency portfolio management. In addition, from the metaverse gamers' perspective, this study aims to assess whether playing continuously for the play-to-earn game to get metaverse-based cryptocurrencies is recompense with the positive returns or not. From the companies' perspective, the optimistic prediction of metaverse-based cryptocurrencies encourages them to develop more metaverse projects. In this study, deep learning methods based on Convolutional neural networks (CNN), Long short-term memory (LSTM), and Gated recurrent units (GRUs) were proposed for predicting the closing prices of Smooth Love Potion (SLP), Sandbox (SAND) and Decentraland (MANA) using 4 different features. The MAPE value of 2.8091 and 3.1813 were achieved by the Previous Close in CNN and Primary Technical Indicator in GRUs models respectively, followed by 4.1331 achieved by the LSTM model using Previous Close feature. A total of 12 test case models were developed based on previous closing price, primary technical indicator and advanced technical indicator and best optimal features for the SAND cryptocurrency prediction and achieved a MAPE error less than 11.0624. The models were also tested with SLP and MANA cryptocurrencies. The best results for SLP are using CNN as model, Primary Technical Indicator(PTI) as features and recorded MAPE value of 153.6376. The best result for MANA is using LSTM as model, Advance Technical Indicator(ATI) as features and recorded MAPE value of 10.8895.

ABSTRAK

RAMALAN MATA WANG KRIPTO BERASASKAN METAVERSE DENGAN MENGGUNAKAN PEMBELAJARAN MESIN.

Mata wang kripto telah menarik perhatian ramai dari pelabur dan penyelidik baru-baru ini. Mata wang kripto telah menjadi fenomena global dalam sektor kewangan dan pilihan keutamaan pelabur untuk instrumen kewangan yang diperdagangkan kerana kemudahannya, inovasinya, keselamatannya, transparansinya, serta sifatnya yang tidak berpusat. Mata wang kripto berasaskan metamesta adalah satu subtopik penggunaan khusus untuk Mata wang kripto yang digunakan untuk perdagangan dan pelaburan dalam aset digital dan tanah di metamesta, dunia maya yang merupakan pengembangan daripada dunia sebenar. Mata wang kripto ini, termasuk yang berasaskan metamesta, mempunyai perubahan yang drastik dan pergerakan yang agresif dalam harga mereka, yang sangat tidak dapat diramal. Walaupun terdapat beberapa kajian terbaru untuk implementasi mesin pembelajaran untuk meramal mata wang kripto, ia tidak pernah dilaksanakan dalam Mata wang kripto berasaskan Metamesta. Kajian ini akan merancang model mesin pembelajaran untuk ramalan harga mata wang kripto berasaskan metamesta. Selain itu, banyak algoritma mesin pembelajaran akan dilaksanakan sebagai model ramalan harga penutup Mata wang kripto berasaskan metamesta. Pembangunan model mesin pembelajaran akan mengikuti proses berikut: pengumpulan data, pra-pemprosesan data, pemilihan model, latihan dan pembangunan model, dan evaluasi model. Akhirnya, sumbangan kajian ini boleh dikesan sebagai berikut. Kajian ini bertujuan untuk membantu pelabur dan penyelidik untuk mengurangkan risiko dalam pasaran cryptocurrency dan menyebarkan pelaburan cryptocurrency. Selain itu, dari perspektif pemain metamesta, kajian ini bertujuan untuk menilai sama ada bermain secara berterusan untuk permainan play-to-earn untuk mendapatkan mata wang kripto berasaskan metamesta adalah sesuai dengan pulangan yang positif atau tidak. Dari perspektif syarikat, ramalan yang optimis tentang mata wang kripto berasaskan metamesta menggalakkan mereka untuk mengembangkan projek metamesta yang lebih banyak.

Dalam laporan ini, kaedah pembelajaran mendalam berdasarkan rangkaian neural konvolusi (CNN), ingatan jangka pendek panjang (LSTM) dan unit berulang berkunci (GRUs) dicadangkan untuk meramal harga penutup Smooth Love Potion (SLP), Sandbox (SAND) dan Decentraland (MANA) menggunakan 4 ciri yang berbeza. Nilai MAPE 2.8091 dan 3.1813 dicapai oleh harga penutup sebelumnya dalam model CNN dan Petunjuk Teknikal Asas dalam model GRUs, diikuti oleh 4.1331 yang dicapai oleh model LSTM menggunakan ciri harga penutup sebelumnya. Sebanyak 12 model kes ujian telah dibangunkan berdasarkan harga penutup sebelumnya, petunjuk teknikal asas dan petunjuk teknikal maju dan ciri-ciri terbaik untuk ramalan matawang kripto SAND dan mencapai ralat MAPE kurang daripada 11.0624. Model-model juga diuji dengan matawang kripto SLP dan MANA. Hasil terbaik untuk SLP adalah menggunakan model CNN, Petunjuk Teknikal Asas (PTI) sebagai ciri dan nilai MAPE 153.6376 dicatat. Hasil terbaik untuk MANA adalah menggunakan model LSTM, Petunjuk Teknikal Maju (ATI) sebagai ciri dan nilai MAPE 10.8895 dicatat.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF APPENDICES

Page

CHAPTER 1

INTRODUCTION

1.1 Introduction

Cryptocurrencies are virtual currencies secured by cryptography (Lobban, 2021). Cryptocurrencies are peer-to-peer currencies because they are decentralised and open-source (Kypriotaki et al., 2015). Cryptocurrencies primarily employ a complicated cryptographic technique: blockchain technologies that necessitate the deployment of a network of computers to perform computationally costly mathematical operations (Choo, 2015). Cryptocurrencies that use a cryptographic architecture can secure transactions because the transaction mechanism is independent of any monetary authority. These currencies are unregulated and highly volatile compared with commodities and stocks due to many factors such as supply and demand, user perceptions, government ordinances, and media hype (Reiff, 2022). As a result, these currencies have drastic fluctuations and aggressive movements in their prices.

Metaverses and play-to-earn games have emerged due to blockchain technology and the gaming industry (Vidal-Tomás, 2022). Play-to-earn games are the game business model where users gain rewards when playing the game. For the significant feature of these games, the users can get rewarded with two types of in-game assets that have intrinsic value. These rewards include the NFT in-game items with diverse scarcity and can be traded and transferred on the open market such as OpenSea, and other rewards are the metaverse cryptocurrencies, cryptocurrencies that can trade and purchase the digital assets in the metaverse. Metaverse is a post-reality universe, a continuous and persistent multiuser environment that integrates physical reality and digital virtuality (Mystakidis, 2022), allowing users to connect using a specific avatar. As the metaverse resembles the real world, it has economic governance and metaverse commerce. Metaverse currencies are currencies in circulation used widely in metaverse commerce, the cornerstone of the economy (Lee et al., 2021) inside the metaverse. The metaverse-based cryptocurrencies have similar features to the traditional cryptocurrencies such as Bitcoin and Etherium, as most of them are built on the Ethereum blockchain network, for example, the Sandbox (SAND) and Decentraland (MANA) (Jeon et al., 2022). The argument can be made as metaverse-based cryptocurrencies extend the typical cryptocurrencies used in the payment method in several metaverses.

Peoples treats metaverse-based cryptocurrencies as the new preferred choice for traded financial instruments. The emergence of metaverse-based cryptocurrencies allows traders to invest in the game and metaverse projects without interacting inside the metaverse, playing and playing-to-earn games (Vidal-Tomás, 2022). Due to its highly volatile price movement influenced by several factors such as supply and demand, user perceptions, government ordinance, and media hype, the investment is considered risky for the investor as the cryptocurrencies quickly devalue overnight. This study proposes numerous machine learning models to analyse the metaverse-based cryptocurrencies and compare which algorithm, parameters, and approach are best suited for metaverse-based cryptocurrencies prediction. The study will be constructed to test the feasibility of predicting extremely volatile metaverse-based cryptocurrencies by using the machine learning method. Since these metaverse-based cryptocurrencies are the newly launched, as most have appeared in the last two years, the study also examines the viability of the short predictive interval towards the time series analysis. The study will also examine the efficiency and accuracy of deep learning to forecast the time series data. Finally, the following is a breakdown of the work's contribution: First, this study can help the investor and researcher to help in minimising the risk in the cryptocurrency market and diversify cryptocurrency portfolio management. In addition, from the gamers' perspective, this study aims to assess whether playing continuously for the play-to-earn game to get metaverse-based cryptocurrencies is recompense with the positive returns or not. From the companies' perspective, the positive performance of the metaverse-based cryptocurrencies will encourage companies' involvement in developing more metaverse projects.

1.2 Problem Background

Peoples treats metaverse-based cryptocurrencies as the new preferred choice for traded financial instruments. The emergence of metaverse-based cryptocurrencies allows traders to invest in the game and metaverse projects without interacting inside the metaverse, playing and playing-to-earn games (Vidal-Tomás, 2022). Due to its highly volatile price movement influenced by several factors such as supply and demand, user perceptions, government ordinance, and media hype, the investment is considered risky for the investor as the cryptocurrencies quickly devalue overnight.

Although there are no existing studies of machine learning regarding the prediction of metaverse-based cryptocurrencies, the literature review will focus on the price prediction of cryptocurrencies. Some recent studies have shed some light on predicting the price and trend of cryptocurrencies. Patel et al.(2020) proposed a hybrid model based on LSTM and GRU that focuses on Litecoin and Monero. The results demonstrate that the proposed models accurately forecast prices with high accuracy and low prediction error, indicating that the scheme applies to numerous cryptocurrencies. Petrovic et al. (2021) proposed a Hybrid Machine Learning and Beetle Antennae Search technique for cryptocurrency price prediction. The results indicate that the CESBAS-ANFIS method outperforms existing approaches such as the LSTM and LSTM-GRU hybrid models in predicting Litecoin and Monero and algorithms for machine learning and compared the models. Chowdhury et al. (2020) proposed a method for predicting and forecasting the closing prices of the cryptocurrency index 30 and nine cryptocurrency constituents using machine learning algorithms. The machine learning model achieved 92.4 percent accuracy using the ensemble learning method and 90 percent accuracy using gradient boosted trees to predict the cryptocurrency index 30 and its nine constituents. Hitam et al. (2019) suggested a Cryptocurrency Forecasting technique based on Particle Swarm Optimization (PSO) and Optimised Support Vector Machine (SVM). The Optimised

SVM-PSO algorithm is preferable to the single SVM algorithm in forecasting the future price of bitcoin. Felizardo et al. (2019) conducted a comparative study on Bitcoin price prediction utilising WaveNets, Recurrent Neural Networks, and machine learning techniques such as ARIMA, SVR, and SVM. The results vary according to the prediction interval; SVM performs best when the prediction interval is 1 and 5 days; ARIMA and SVR perform best when the prediction interval is 10 and 30 days, and LSTM and WaveNet perform best when the prediction interval is 30 days. Rathan et al. (2019) proposed a technique for forecasting Crypto-Currency prices through Decision Tree and Regression approaches. The results demonstrate that linear regression is more efficient at predicting bitcoin prices than decision trees, with an accuracy of 97.5 percent versus 95.8 percent. Derbentsev et al. (2020) forecast bitcoin values using an ensembles-based machine learning approach. The results indicated that using ensemble tree-based models such as GBM and RF for short-term forecasting of cryptocurrency time series is efficient, with GBM and RF predicting the Ripple price by 0.92 percent and 1.84 percent, respectively. Phaladisailoed and Numnonda (2018) compared different machine learning models for bitcoin price prediction, including Theil-Sen regression, LSTM, Huber regression, and GRUs. The results indicated that GRU outperformed the other three approaches, with a Mean Squared Error (MSE) of 0.00002 and an R square of 99.2 percent. Indulkar (2021) proposed a time series analysis of cryptocurrencies like Bitcoin, Ethereum, Chainlink, Bitcoin Cash, and Ripple using Deep Learning and Fbprophet over a range of time frames. The results indicated that the Bitcoin cryptocurrency generated the fewest errors at 0.01867, followed by Bitcoin Cash at 0.02632.

Therefore, in this study, multiple machine learning models are proposed to analyse the metaverse-based cryptocurrencies and compare which algorithm, parameters, and approach are best suited for predicting metaverse-based cryptocurrencies. The study will be constructed to test the feasibility of predicting extremely volatile metaverse-based cryptocurrencies by using the machine learning method. Since these metaverse-based cryptocurrencies are the newly launched cryptocurrencies, as most of them appeared in the last two years, the study also examines the viability of the short predictive interval towards the time series analysis. The study will also examine the efficiency and accuracy of deep learning to forecast the time series data. Finally, the following is a breakdown of the work's contribution: First, this study can help the investor and researcher to help in minimising the risk in the cryptocurrency market and diversify cryptocurrency portfolio management. In addition, from the metaverse gamers' perspective, this study aims to assess whether playing continuously for the play-to-earn game to get metaverse-based cryptocurrencies is recompense with the positive returns or not. From the companies' perspective, the positive performance of the metaverse-based cryptocurrencies will encourage companies' involvement to develop more metaverse project.

1.3 Problem Statements

In the few years, many papers published have been using deep learning and machine learning to predict the price of cryptocurrencies. However, it has never been implemented in Metaverse-based cryptocurrencies. The research shows that the price movements of metaverse cryptocurrencies are not related to the traditional cryptocurrencies market trend (Vidal-Tomás, 2022), and the metaverse cryptocurrencies move more vigorously than traditional ones. The Decentraland (MANA) prize movement surged more than 4500 % in 2021, while the Bitcoin was recorded at 200%(Noonan, K., 2021) means metaverse cryptocurrencies is more volatile than the traditional cryptocurrencies. The challenge of this study is to accurately predict the future closing price of the given Metaverse-based cryptocurrencies across a given time frame in the future. For this study, the different machine learning algorithms will apply to predict the closing price of Smooth Love Potion (SLP), Sandbox (SAND), and Decentraland (MANA) by using the multiple features of datasets.

1.4 Project Objectives

The following are the few objectives that must be accomplished in order for the study's goal to be achieved:

1. To curate and modify the existing metaverse-based cryptocurrencies' prices datasets and examine the performance and efficiency of using different features of datasets to forecast metaverse-based cryptocurrencies' prices.
2. To design and implement the Convolutional neural networks (CNN), Long short-term memory (LSTM), and Gated recurrent units (GRUs) machine learning algorithms in the predictive models to forecast metaverse-based cryptocurrencies' prices.
3. To evaluate the performance and efficiency of the machine learning models by using Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE).

1.5 Project Scope

This project's scope is to predict the metaverse cryptocurrencies such as SAND, SLP and MANA. Based on the dataset, the prediction timeframe is only focused on the last 10 days of the metaverse cryptocurrencies such as SAND, SLP and MANA. The algorithms that are used for the model building are only limited to the deep learning algorithms, which are Long short-term memory (LSTM), Gated recurrent units (GRUs) and Convolutional neural network (CNN).

1.6 Organization of the Report

After the introductory section, chapters 2 describe the systematic literature review of the cryptocurrency by using machine learning. The following section will summarise the dataset, machine learning approaches used, interval predicted, and types of cryptocurrencies predicted. This section also categorizes the literature review based on the machine learning approach, published year, and country of origin. Chapter 3 details the methodology applied in this project with five main stages of the model building process: data collection, data pre-processing, model choosing, model training and development, and model evaluation. Chapter 4 describes the data and feature engineering used in experiments 1 to 20, the model design and architecture of the proposed LSTM, GRUs and CNN model, the model evaluation metrics and the setup of the experiment, including the parameter setup dependent and target variables of both 20 experiments. Chapter 5 describes the python code implementation of experiments 1 to 20 using Google Colab as the platform for SAND, SLP and MANA metaverse cryptocurrencies 10 days future price prediction by using the proposed LSTM, GRUs and CNN model based on the different input features. Chapter 6 discusses the results of experiments 1 to 20 to three metaverse cryptocurrencies: SAND, SLP and MANA, in the metrics of MSE, MAE, RMSE, and MAPE. Chapter 7 provides an overview of the project's summary, objectives and achievements, main findings, limitations and potential for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Cryptocurrencies are a type of virtual currency secured by cryptography (Lobban, 2021). Cryptocurrencies are peer-to-peer currencies because they are decentralized and open-source (Kypriotaki, 2015). Cryptocurrencies primarily employ a complicated cryptographic technique that necessitates the deployment of a network of computers to perform computationally costly mathematical operations (Choo, 2015). Cryptocurrencies that use a cryptographic architecture can secure transactions because the transaction mechanism is independent of any monetary authority. However, these currencies are unregulated and extremely volatile compared with commodities and stocks due to many factors such as supply and demand, user perceptions, government ordinances, and media hype (Derbentsev et al., 2020). As a result, these currencies have drastic fluctuations and aggressive movements in their prices.

While the prices of cryptocurrencies go up drastically and break the multiple new all-time high prices in 2021, people's enthusiasm, even the companies such as Tesla, to invest in cryptocurrencies, has skyrocketed. The development of Metaverse, Non-fungible token (NFT), and Decentralized finance(Defi) also enhances the use of cryptocurrencies not only apply at commercial transactions and transfer of assets in the real words in the virtual world. Furthermore, the universal use cases of cryptocurrencies cause numerous physical approaches and modelling techniques to model and analyze the price of cryptocurrencies.

In this literature review, the use of machine learning and deep learning techniques to predict cryptocurrency prices is presented and summarized. Through this review, common approaches, analysis types, and findings were identified. The current state of research in this field was evaluated, and future research possibilities were identified. By providing a summary of previous work, identifying recurring trends and unfilled niches, this literature review makes a three-fold contribution. Firstly, it provides scholars in this field with a comprehensive overview of previous research, which can help guide future studies. Secondly, it highlights promising strategies for solving the cryptocurrency price prediction problem. Thirdly, it establishes reporting guidelines to improve transparency and accelerate scientific progress in this field.

Figure 1 below shows the cryptocurrency prediction system frameworks based on different algorithms and approaches.

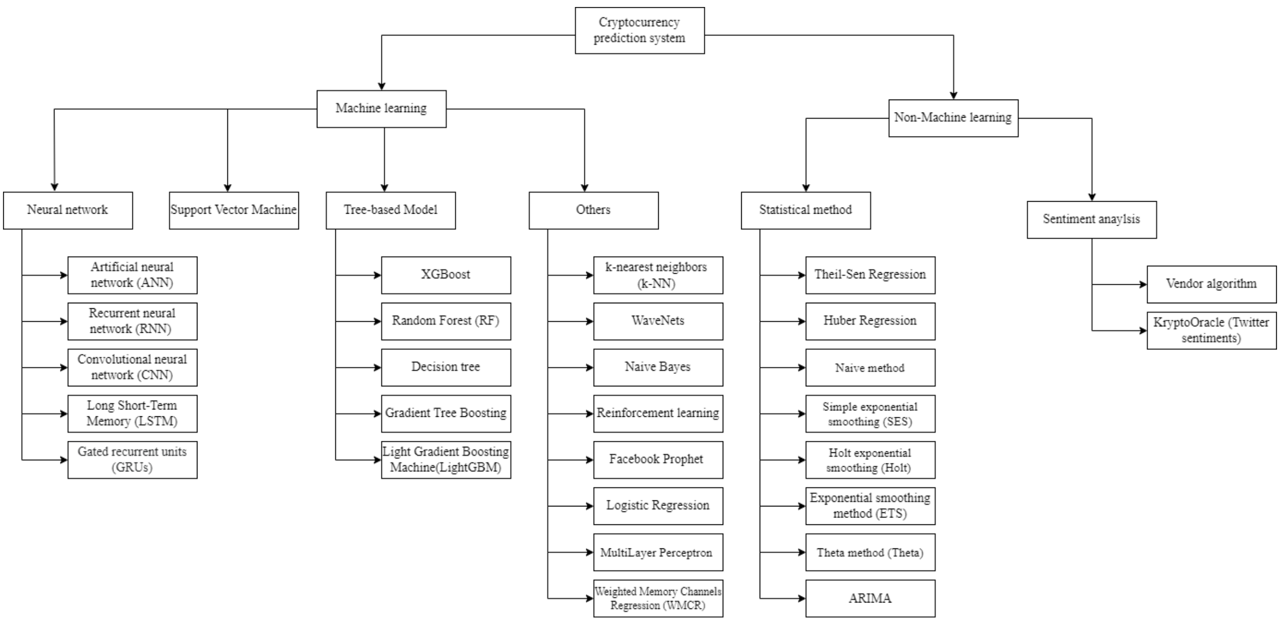


Figure 1: Overview of Cryptocurrency Prediction Systems

2.2 Methodology of Literature Review

A literature search was conducted based on the recommendation of Webster and Watson (2002) and Vom Brocke et al. (2009). Many interdisciplinary research databases were examined to establish an initial literature base, including the ACM Digital Library, Emerald insight, IEEE, SpringerLink, ScienceDirect, and Scopus. The Google Scholar, Scopus, and IEEE databases were then submitted with basic machine learning and cryptocurrency keywords, and the scope of the search topic was limited to recent years, specifically those greater than 2017. Queries were submitted as follows:

Google Scholar: allintitle: "machine learning cryptocurrency," resulting in 157 documents.

Science Direct: (TITLE-ABS-KEY (machine AND learning) AND TITLE-ABS-KEY (cryptocurrency)), resulting in 273 documents.

IEEE: ("All Metadata": machine learning) AND ("All Metadata": cryptocurrency), resulting in 201 documents.

Springer: "cryptocurrency AND machine AND learning," exclude the Preview-Only content resulting in 180 documents.

A preliminary list of 66 publications was obtained, which were then further reviewed. Out of these, 51 papers were eliminated based on their titles and abstracts as they did not expressly meet the subject of the literature review. This could be due to the papers implementing methods that did not meet Gu et al.'s (2019) definition of machine learning, papers that did not focus on cryptocurrency price and return prediction, papers that were not available in English, or papers not using a prediction task, papers that only focused on statistical methods and did not include machine learning scope. A forward and backward search was then implemented for the remaining relevant papers, resulting in another 15 articles, bringing the total number of publications for an in-depth review to 30.

A classification of the reviewed literature will be scrutinized and categorized by country of origin, year of publication, and source of literature obtained. To further analysed the methodology and approaches used by the reviewed literature, fundamental concepts for categorizing the price prediction techniques within all review literature will be established. An initial set of classification concepts will be gone over and developed. These early notions will be reviewed throughout the paper screening process and adapted as needed. Following that, all of the identified concepts were discussed and synthesised, yielding a final set of categorisation concepts:

- Machine learning (Recurrent Neural networks method, Support Vector Machine method, SVM method, Tree-based method and Others)
- Features (i.e., technical-base, sentiment-based, asset-based and blockchain-based)
- Predictive intervals (i.e., second, minute, hour, day, week)
- Cryptocurrency type (i.e., Bitcoin, Dash, DOGE, Ethereum, IOIA, Litecoin, NEM, NEO and so on)

2.3 Classification of the Reviewed Literature

In all, 30 peer-reviewed research publications were analysed and classified according to country of origin, publication year, and source of literature obtained; this part shows the classification of the scrutinised papers.

2.3.1 By Country

Figure 2 below illustrates the distribution of reviewed articles according to the country in where the study was undertaken. This area piqued the interest of researchers from a wide variety of countries. The research articles that were analysed were published between 2018 and 2022 and were from 18 different nations. India submitted the most papers (8) for review. An assessment of four articles from each of the following countries was carried out: the United States, Brazil, China, Malaysia, South Korea, and the United Kingdom; and two papers from each of the following countries were also assessed: Bangladesh, Canada, Dubai, Greece, Portugal, Serbia, Switzerland, Thailand, Turkey, and Ukraine.

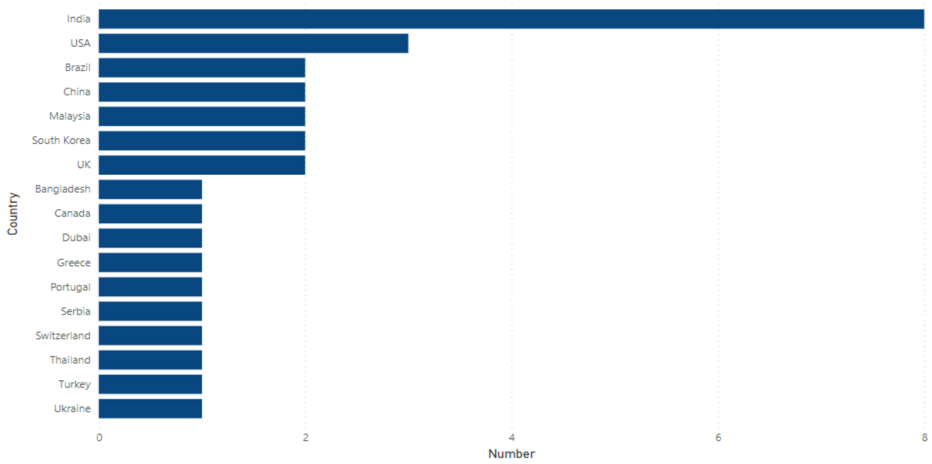


Figure 2: Classification of Studies on Cryptocurrency Market Forecasting by Country

2.3.2 By Year

According to the articles picked, a year-by-year evaluation of the papers is illustrated in Figure 3. It was discovered that studies on the cryptocurrency market have surged in recent years. The results demonstrate that research interest in this topic has increased. All peer-reviewed publications occurred within the last five years: 2021 (12), 2020 (9), 2019 (7), and 2018 (2) were recorded.

Figure 3: Year-wise Categorization of Studies on the Cryptocurrency Market

2.3.3 Source Based

Relevant papers for cryptocurrency price prediction that received at least one citation published between 2018 and 2021 on Elsevier, Springer, or IEEE Xplore were chosen. In Figure 4, the distribution of publications by the journal can be examined. According to the articles collected from these publications, nearly half (17) of the papers evaluated were from IEEE Xplore. Elsevier published ten of the papers evaluated, while Springer published five.

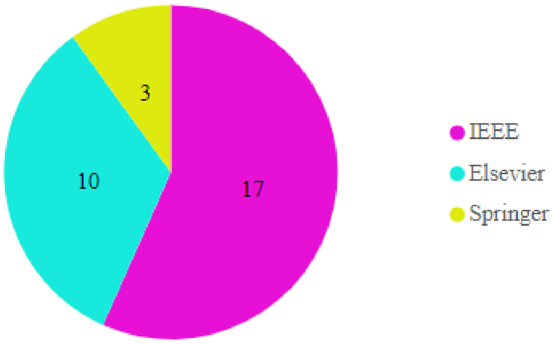


Figure 4: Distribution of Papers According to the Journals

2.4 Brief Review of Cryptocurrency Prediction Techniques

To assess and structure the literature, four main concepts were employed: predictive features, prediction intervals, machine learning approaches, and type of cryptocurrencies forecasted (Jaquart et al., 2020). These principles are broad and can be applied to various prediction problems. Some concept features, such as blockchain-based features, are unique to the crypto pricing problem. It is impossible to compare the models across articles since they study distinct periods, different parameters, and methods, and are evaluated using separate evaluation metrics. However, because they all use the same data, comparing multiple machine learning models within the same paper is still viable. Even within the same paper, however, model comparisons are only valid if all models are equally ideally tuned and then used predictive interval reflects cryptocurrency's price formation process.

2.4.1 Machine Learning Techniques

The body of literature studied used a variety of different machine learning techniques. Based on the introduced models, we classify the literature into five categories. Several types of neural networks exist, such as recurrent neural networks, tree-based models, support vector machines, and multiple models.

The recurrent neural network is a sort of neural network in which the previous state's output is used as feedback for the current state, similar to the regular neural networks in which the input and output are independent of each other. There is only one input layer, some hidden layers, and an output layer in feedforward neural networks (Shi et al., 2017). The RNN's hidden layer is made up of a loop that feeds back on the current state. As a result, the state of the hidden layer at any given time is influenced by both previous and present input (Yang et al., 2016). Gated recurrent units (GRUs), Artificial neural networks (ANN) and Long short-term memory (LSTM) are example of recurrent neural networks.

Support vector machines (SVMs) are supervised learning models that evaluate data for classification and regression analysis. Support vector machines are based on the principle of minimising generalisation error by generating a (set of) hyperplane (s) in a high-dimensional space.

Tree-based model is a subset of supervised Machine Learning models that conducts classification and regression tasks by constructing a tree-like structure for classifying or valuing the target variable based on its features. In tree-based models, the outcomes are cuboid regions with axis-aligned edges (Crosby et al., 2016). A frequently used methodology implementation is the random forest, which constitutes an ensemble of imperfectly correlated trees to reduce the variance of forecasts.

Multiple models are the integration of two of the models stated above to solve the research question in the articles, such as using various multiple machine learning algorithms to predict the cryptocurrency returns and prices. Others employ methods that are part of none of the four major categories above.

Recurrent Neural Networks

Patel et al. (2020) suggested a hybrid cryptocurrency prediction system based on GRU and LSTM that was limited to Monero and Litecoin. The results demonstrate that the suggested scheme accurately forecasts prices with high precision, implying that it can be used for a broad range of cryptocurrencies. The findings demonstrate that the suggested technique accurately forecasts the Litecoin and Monero cryptocurrency prices across the specified window sizes of 1, 3, and 7 days. The LSTM-GRU hybrid model's proposed technique surpassed the LSTM network by the lower MAPE, MAE, RMSE, and MSE values.

Petrovic et al. (2021) proposed a hyper-parameter optimization system based on the architecture of the adaptive neuro-fuzzy inference system (ANFIS) and the hybrid algorithm of the Cauchy exploration strategy beetle antennae search (CESBAS). ANFIS is a technique for the fuzzy inference that is part of the family of artificial intelligence techniques. The study's findings indicate that the proposed model beats other recent similar strategies in terms of accuracy and can be used successfully for this critical task. The CESBAS-ANFIS model has a lower MSE, RMSE, MAE, and MAPE than the LSTM-GRU models and LSTM models.

Yiying et al. (2019) developed advanced artificial intelligence frameworks of fully connected Long Short-Term Memory (LSTM) and Artificial Neural Network (ANN) to analyse the price movements of Ethereum, Ripple, and Bitcoin. They discovered that ANN relies more on long-term history than LSTM, meaning that LSTM is more efficient at extracting useful information from historical memory than ANN. However, when sufficient historical data is available, ANN can achieve similar accuracy to LSTM. Bitcoin and Ripple perform well when the time interval is one day, whereas Ethereum performs well when the prediction period is three days. Ethereum and Ripple perform well when the time interval is 7 days, whereas Bitcoin performs well when the prediction period is 14 days.

Multiple-Models

Chowdhury et al. (2020) suggested a method that uses machine learning models and algorithms to anticipate and forecast the closing price of the cryptocurrency index 30 and its nine constituents, thereby making it easier for consumers to trade these currencies. They achieved 92.4 percent accuracy with ensemble learning, and 90 percent accuracy using gradient boosted trees, with RMSE values of 0.001 and 0.002, respectively.

Felizardo et al. (2019) compared multiple machine learning techniques for predicting the future price of Bitcoin, including Random Forest (RF), ARIMA, Long Short-Term Memory (LSTM), WaveNets, and Support Vector Machine (SVM). For prediction interval, which is 1 day and 5 days, SVM is the best performing model in MSE, MAE, MAPE, RMSE, and MPE evaluation metrics. ARIMA and SVR are the best models for ten days prediction intervals. Finally, for the 30days prediction interval, WaveNet or the LSTM performs better than other models.

Borges et al. (2020) describe a machine learning-based approach for constructing a cryptocurrency trading strategy. Additionally, rather than investing in predictions based on time-sampled financial series, this work developed and applied a novel method for resampling financial series to generate investments with higher returns and lower risk. For this reason, the original time-sampled financial series are resampled using a closing value threshold, resulting in a series with greater returns and lower risk than the original. Technical indicators are created and fed into four machine learning algorithms: Gradient Tree Boosting, Random Forest, Logistic Regression, Support Vector Classifier, and resampled and original data. Each of these algorithms is responsible for the development of a transaction signal. To increase the performance of the previous algorithms, a fifth transaction signal is created by calculating the unweighted average of the four trade signals they generate. Finally, the investment outcomes of the resampled series are compared to those of the more often employed fixed time interval sampling. This study demonstrates that regardless of whether resampling is employed, all learning algorithms outperform the Buy and Hold (B&H) strategy in the vast majority the of the 100 markets analysed. Nonetheless, the unweighted average outperforms all other learning algorithms, with accuracies of up to 59.26 percent for time-resampled series.

Zhang et al. (2021) propose a Weighted & Attentive Memory Channels model to forecast cryptocurrencies' daily close price and volatility. Their proposed model is composed of three modules:

- Attentive Memory module: Combines a Gated Recurrent Unit with a self-attention component to establish attentive memory for each input sequence.
- Channel-wise Weighting module: Receives the prices of several major cryptocurrencies and learns their interdependence by recalibrating the weights for each sequence.
- Convolution & Pooling module: Extracts local temporal features to improve generalizability.

A series of tests are conducted to validate the proposed model. The researchers' findings indicate that their suggested system beats baseline models in the accuracy, profitability, and prediction error. WAMC forecasts the Ethereum market with an accuracy of 77.52 percent and the Bitcoin Cash market with an accuracy of 75.23 percent.

The research was conducted by Kim et al. (2021) to look into the relationship between Ethereum Blockchain information and Ethereum pricing. Additionally, investigate at how Ethereum values are related to Blockchain information about other publicly accessible currencies on the market. Their significant findings show that macroeconomic variables, Ethereum-specific Blockchain information, and Blockchain information from other cryptocurrencies all play a role in Ethereum price prediction. In every model, ANN outperforms SVM in terms of accuracy. The ANN is used in Models I-4 (RMSE=0.068) and II-4 (RMSE=0.068) to record the maximum accuracy with the lowest RMSE value.

Akyildirim et al. (2020) investigate the daily and minute-by-minute prediction of the twelve most liquid cryptocurrencies by integrating machine learning classification techniques such as logistic regression, support vector machines, random forests, and artificial neural networks with historical price data and technical indicators as model components. The average classification accuracy of four algorithms is consistently greater than 50% for all cryptocurrencies and timelines, demonstrating that price trends in the cryptocurrency markets can be forecast to some extent. On a daily or minute-by-minute basis, machine learning classification algorithms achieve an average prediction accuracy of approximately 55–65 percent, with support vector machines achieving the highest and most consistent predictive accuracy compared to logistic regression, support vector machines, random forests, and artificial neural networks classification algorithms.

Phaladisailoed et al. (2018) identify the most efficient and accurate model for predicting Bitcoin values among various machine learning methods. Various regression models with scikitlearn and Keras libraries were tested using 1-minute interval trading data on the Bitcoin exchange website Bitstamp from January 1, 2012, to January 8, 2018. The Mean Squared Error (MSE) was as low as 0.00002, and the R-Square (R^2) was as high as 99.2 percent in the top findings.

Yogeshwaran et al. (2019) used the necessary quantity of data and processing power to build a machine model to predict the price of a cryptocurrency. When different layers of CNN are tested, the outcome shows that the three-layer technique outperforms the other two models. The four-layer strategy comes next, followed by the two-layer approach.

Samaddar et al. (2021) compared the results of numerous machine learning models, including graphs of epoch versus error, accuracy, and price for each model that used both linear and non-linear functions. They conducted the study using both neural network approaches such as recurrent neural networks (RNN), artificial neural networks (ANN), and convolutional neural networks (CNN), as well as supervised learning algorithms such as k-nearest neighbours (k-NN) and Random Forest (RF). CNN has the highest accuracy (99.7%) compared to other algorithms because it obtains the highest accuracy and has the lowest loss (0.000162046).

Politis et al. (2021) proposed an approach for creating deep learning models for forecasting cryptocurrency values and applying it to the Ethereum price prediction, yielding short- and long-term predictions with an accuracy of up to 84.2 percent. They employed the Mean Absolute Performance Error (MAPE) and Root Mean Square Error (RMSE) for the regression problem, but mainly the accuracy for the classification problem. They constructed ensemble models by integrating our models' predictions with those stated above. All models performed wonderfully in both the regression and classification problems. The Hybrid LSTM-GRU model performed the best in regression-creating predictions, with an RMSE of 8.6 and a MAPE of 3.6 percent for daily forecasts. With an accuracy of 84.2 percent, the Ensemble model outperformed all other models in the classification test. In this case, the optimal Ensemble model includes the predictions of the LSTM, Hybrid LSTM-GRU, and Hybrid LSTM-TCN models.

Akyildirim et al. (2021) examined the performance of various types of machine learning algorithms (MLAs) in predicting mid-price movement for Bitcoin futures pricing. They examine the relative forecasting performance throughout a range of temporal frequencies, from 5 to 60 minutes, utilising intraday high-frequency data. Their findings indicate that five of the six MLAs consistently beat benchmark models such as random walk and ARIMA when forecasting Bitcoin futures prices, indicating that MLAs outperform benchmark models such as random walk and ARIMA.

Freeda et al. (2021) suggested a deep learning method to forecast bitcoin values using a Recurrent Neural Network model that utilizes time-series data to increase accuracy. The work is novel in that it trains and tests the recurrent neural network model on an existing dataset in order to obtain a long-term forecast. This analysis forecasts the value of bitcoin in 2021. Compared to other machine learning algorithms such as Gaussian Nave Bayes, Random Forest, K-Nearest Neighbors, and Support Vector Machine, the proposed work demonstrates that the RNN model improves accuracy by 76.99 percent.

E. Jakubowicz and E. Abdelfattah (2021) presented the study to determine whether increased accuracy may be achieved by focusing on numeric ranges rather than precise time-series price predictions. The forecasts concentrated on the market's predicted trajectory during the next hour. Five different machine learning models were trained and tested using one-hour interval trade data to establish discrete classes of hourly change levels. Except for one model, cross-validation accuracy ranged between 96 and 100%.

Support Vector Machines

N. A. Hitam, A. R. Ismail, and F. Saeed developed PSO to improve SVM algorithms for bitcoin forecasting in 2019. This classifier is tested using a selection of cryptocurrencies. The experimental result indicates that while estimating the future price of bitcoin, an optimised SVM-PSO method outperforms single SVM algorithms. According to the data, SVM-PSO surpassed other classifiers with a 97 percent accuracy. It then determines whether the population size and quality of the training dataset affect the predicting accuracy.

Peng et al. (2018) analyse the volatility prediction performance of three cryptocurrencies and three currencies with recognised stores of value using daily and hourly frequency data. Researchers combined the standard GARCH model with a machine learning method for volatility estimates, estimating the volatility and mean equations with Support Vector Regression (SVR) and comparing them to GARCH family models. Additionally, Hansen's Model Confidence Set and Diebold-Mariano test were employed to evaluate the predictive performance of the models. The same analysis was performed on both low and high-frequency data. According to the results, SVR-GARCH models outperformed EGARCH, GARCH, and GJR-GARCH models when Student's t, Skewed Student's t, and Normal distributions were used. For all variables and both time frequencies, the SVR-GARCH model demonstrated statistical significance favoring its superiority to GARCH and its expansions.

Tree-based Method

Sun et al. (2020) proposed the Light Gradient Boosting Machine (LightGBM), a novel Gradient Boosting Decision Tree (GBDT) technique for estimating the cryptocurrency market's price trend (falling or not falling). They combine daily data from 42 different types of significant cryptocurrencies with critical economic aspects to derive market data. In terms of resilience, the LightGBM model surpasses the other approaches, and the overall strength of the cryptocurrencies affects forecasting performance. When the test set is a subset of the training set or is independent of the training set, RF, LightGBM, and SVM models perform better in 2 weeks than in 2 days and 2 months prediction intervals.

Derbentsev et al.(2020) used machine learning to address the problem of forecasting short-term cryptocurrency time series. Two of the most potent ensemble techniques are Random Forests (RF) and Gradient Boosting Machines (GBM). They validated the models by comparing them to the daily close prices of three prominent coins: Ethereum (ETH), Ripple (XRP), and Bitcoin (BTC), as well as historical price data and moving average. The researchers employed a one-step forward technique to assess the models' performance to generate out-of-sample projections for three cryptocurrencies. The accuracy rate for the models was determined using the Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) measures. According to a comparison of the predictive abilities of the RF and GBM, both models predicted out-of-sample datasets with comparable accuracy, whereas boosting was somewhat more accurate.

Rathan et al. (2019) determined the price trend based on day-to-day variations in the Bitcoin price. The dataset includes open, close, low, high, and price details for Bitcoin value up to the present day. This research aims to calculate and compare the accuracy of Bitcoin prediction using several machine learning algorithms. The results of the decision tree and regression model experiments are compared. The proposed learning approach suggests the optimal algorithm for the cryptocurrency prediction problem to choose and implement. The results of the experimental study reveal that linear regression outperforms the other in terms of price prediction accuracy.

Others

Mohapatra et al. (2019) introduced KryptoOracle, a ground-breaking real-time and adaptable cryptocurrency price prediction system based on Twitter sentiments. The platform's integrative and modular architecture is based on three components: a Spark-based architecture capable of persistently and fault-tolerantly handling large volumes of incoming data; a sentiment analysis approach capable of responding in real-time to large volumes of natural language processing queries; and a predictive method based on online learning in which a model adjusts its weights to account for new prices and sentiments. Additionally, the study discusses the development and experimental evaluation of the KryptoOracle platform and provides an architectural design. This approach is reported to be accurate to within a root mean square error (RMSE) of ten dollars between actual and expected Bitcoin values.

Kraaijeveld et al. (2020) investigated Twitter sentiment's predictive power in predicting price returns for the nine most popular cryptocurrencies: Ethereum, Bitcoin, Ripple, EOS, Bitcoin Cash, Cardano, Litecoin, TRON, and Stellar. It was discovered that Twitter sentiment has predictive power for the returns of Litecoin, Bitcoin, and Bitcoin Cash using a cryptocurrency-specific lexicon-based sentiment analysis approach, financial data, and bilateral Granger causality testing. Predictive power for EOS and TRON is discovered using a bullishness ratio. Finally, a heuristic approach is devised to determine that at least 1–14% of the Tweets received were posted by Twitter "bot" accounts. This will be the first paper to look into the predictive capacity of Twitter sentiment in the context of several cryptocurrencies and the presence of cryptocurrency-related Twitter bots. The results show that all nine cryptocurrencies' polarity scores have remained steady over time. With a mean polarity of 0.33, the scores are also consistently positive biassed. Predictive power for price return is observed when predicting TRON and EOS when evaluation metrics bullishness ratio is used.

Gidea et al. (2018) evaluated four major cryptocurrencies: Bitcoin, Ethereum, Ripple, and Litecoin, prior to the onset of 2018's digital asset market crash. Additionally, they examine Bitcoin prior to several of the mini-crash events between 2016 and 2018. All relevant time series behaved erratically. To identify the emerging chaotic regime in a complex system on the verge of a critical shift, the researchers propose a methodology that combines topological data analysis and k-means clustering. They initially tested their methods on the system dynamics of a Lorenz-complex type attractor. Then they applied it to the four most widely used cryptocurrencies. Researchers see early warning indications of key shifts, such as collapses, with a 75 percent accuracy rate in the cryptocurrency markets, as six out of eight datasets record true positives, while the other two record false positives.

In light of the datasets' seasonality, Y. Indulkar (2021) designed the study to interpret the time series analysis of multiple cryptocurrencies. This research employs the LSTM approach but also the Fbprophet technique. The MAE metric was utilized to evaluate the performance between the models used; this score was calculated for all cryptocurrencies and indicated that the lower the value, the better the validation dataset prediction. The experiment indicated that Bitcoin generated the fewest errors with a score of 0.01867, followed by Bitcoin Cash with a value of 0.02632. Due to the lower MAE score, both of the cryptocurrencies mentioned above have a similar plot comparing actual data points to projected data points. The result demonstrates that the trained model accurately predicted both digital currencies, as its error rate was low in contrast to others. The other prediction approach used was the Fbprophet model, which is a machine learning model that is used to foresee and understand trend changes based on seasonality, which was daily in this study. This graphic shows the 15-day prognosis for each currency.

Waheeb et al. (2020) compare statistically and machine learning approaches to predicting Bitcoin's closing values. Thirteen forecasting techniques were used to forecast the closing price of the Bitcoin for the next 14 days. The methods that are used are naive, average, auto-regressive integrated moving-average, drift, Holt, simple exponential smoothing (SES), and damped exponential smoothing, the average of SES, Holt, and damped exponential smoothing, exponential smoothing (ETS), Theta, bagged ETS, multilayer perceptron, and extreme learning machines (ELM). The findings of the investigation are triangular. To begin, seven forecasting algorithms, including ELM, MLP, basic exponential smoothing, damped exponential smoothing, ETS, and Theta, outperformed the naive technique. Second, MLP and ELM forecasted more accurately than the other forecasting methods used in this study on both out-of-sample and validation data. Third, training data is crucial and should be factored into forecasting system development.

Shahbazi et al. (2021) proposed a machine learning-based solution to the price prediction of Bitcoin for a financial institution. The proposed system is built on a blockchain to maintain a secure transaction environment, and it makes use of a reinforcement learning algorithm for price analysis and prediction. This method is primarily applicable to Litecoin and Monero. The results indicate that the proposed system outperforms another state-of-the-art method in terms of price prediction accuracy. The proposed technique dramatically reduces the root mean square error of 3-, 7-, and 30-day Litecoin and Monero price predictions. The most significant gain in Litecoin is from 14.0572 to 3.3097, while the most significant increase in Monero is from 16.1076 to 4.3826.

A comparison of the findings obtained by Samaddar et al. (2021) using various machine learning models is conducted, and graphs for epoch versus accuracy, error, and price for each model using both linear and non-linear functions. To conduct the analysis, they combine neural network methods such as recurrent neural network (RNN), convolutional neural network (CNN), and artificial neural network (ANN) with some well-known supervised learning algorithms such as k-nearest neighbours (k-NN) and Random Forest (RF). The temporal price prediction graphs and the epoch loss accuracy graphs analyse how each algorithm performs differently on the same data. CNN's accuracy rate is almost 99.7 percent, indicating that it made an excellent prediction and barely lost 0.000162046. The accuracy of RF is 99.9957. Although the open value was used as a training sample, it is not a compatible algorithm due to its approximately 0.55797 loss. Though less accurate than both CNN and RNN, ANN nevertheless predicted reasonably well, with a loss of approximately 0.0740, greater than the loss for both CNN and RNN. The k-NN model predicted the data accurately, with a precision of 100.02331 percent, but with a loss of 0.6374, similar to the Random Forest model.

Tiwari et al. (2021) used machine learning techniques to forecast Bitcoin price movements and pricing. They propose to forecast prices using ARIMA, Facebook Prophet, and XGBoost methodologies. The ARIMA, FB Prophet, and XGBoost have a root mean square error of 1165.14, 1300.81, and 13356.43, respectively. ARIMA is the most accurate machine learning technique for forecasting the bitcoin price based on the previous day's pricing.

Rane et al. (2019) conducted in-depth research into the evolution of Bitcoin and a thorough analysis of various machine learning methods for price prediction. The comparative analysis aims to find the best method for forecasting prices more accurately. Non-linear Auto-Regressive with Exogenous Input Model reaches the highest accuracy with 62%. The Regression model (56%) has the best accuracy among the other models. Then the Multilayer perceptron neural network model

(55%), the Support Vector Machine model (54%), the ARIMA model (53%), and the LSTM model (52%) came in second and third, respectively (50%).

2.4.2 Return-predictive Features

A variety of return forecasting signals are used in the literature on cryptocurrency pricing via machine learning. Technical features such as price trends, chart patterns, volume, and momentum indicators are used in the literature on pricing traditional financial products such as bonds, stocks, options, communities, and forex. However, blockchain-based features like Total Hash Rate and Network Difficulty (Kraft, 2016) are unique to cryptocurrencies, particularly Bitcoin and Ethereum. Cryptocurrency and other traditional products like stock, bonds, forex, and commodities have similarities in the trading and order-book mechanism. However, the difference is that cryptocurrencies do have physical use-cases, are not guaranteed by the proper monetary authority and central bank (Wüst & Gervais, 2018), and are not backed up with tangible assets because they do not meet the intrinsic value criteria in the traditional senses. These differences cause the feature categorization for the cryptocurrency need to vary from the other financial assets. Therefore, we divide return prediction variables into four categories based on the studied literature: technical-based, blockchain-based, sentiment and interest-based, and asset-based.

Technical features represent the historical data from the cryptocurrency market, such as the historical open and closing price and trade volumes. Therefore, technical features are the most commonly utilised in the examined literature. Sentiment features represent the social media users' sentiments, such as Twitter, Reddit, and Facebook sentiment, and the number of top search queries on the search engines such as Google Trend data. Asset-based features represent the performance and return of significant world market indices such as Nasdaq Composite, Standard and Poor's 500, Dow Jones Industrial Average, commodity prices such as Brent oil and Gold, and the forex exchange rate and interest of the bond. Blockchain-based features represent the factors influenced by Blockchains, such as total Hash rate and network difficulty.

2.4.3 Interval of Prediction

To forecast the price of cryptocurrencies, the researcher in the studied literature employs a variety of prediction intervals. As a result, the phrase "prediction interval" represents the time stamp between the two prediction values. The prediction intervals range from five seconds to one week in the studied literature. Therefore, the models are divided into five categories based on the prediction intervals: minute, hour, day, week, and month.

2.4.4 Type of Cryptocurrencies Predicted

The authors in the studied literature employ different types of cryptocurrencies to anticipate their results and outcomes. Bitcoin, Ethereum, Litecoin, Dash, Ripple, Zcash, and other cryptocurrencies are widely used. We list the leading cryptocurrency Bitcoin, Ethereum, Ripple, and Zcash are the currency available in the literature review. These are the most commonly used cryptocurrencies, which have a high level of liquidity and are well-known.

Bitcoin is the first and most extensively utilised cryptocurrency in the world. It was established in 2009. It is the first decentralised cryptocurrency that enables transactions via blockchain technology (Crosby et al., 2016). Ethereum is a cryptocurrency network powered by blockchain technology that enables smart contracts. It is a decentralised software that enables the development and deployment of smart contracts on its network without the requirement for third-party control or the risk of fraud (Alharby, M., & Van Moorsel, 2017). Litecoin was founded in 2011 by Charlie Lee, an MIT graduate and Google engineer. It applied Bitcoin technology as its implementation and design (Madakam & Kollu, 2020). Ripple is a real-time gross settlement system, currency exchange, and remittance network developed by Ripple Labs Inc, an American technology startup (Pilkington, 2016). Zcash is a cryptocurrency that use an encryption technique that aims to provide more privacy than other cryptocurrencies like Bitcoin, which launched in 2016 (Kappos et al., 2018). Finally, Dash is a cryptocurrency that is open source. It is a cryptocurrency created as a branch of Bitcoin technology, which was introduced in 2014 (Abdulmonem et al., 2020).

The full details description of the literature review will be provided in the table below based on the source, machine-learning technique, cryptocurrency, prediction interval, and performance.

Table 1: Summary of Literature Review

Scource	Machine-learning technique	Cryptocurrency	Feature	Interval	Performance
Patel et al.	LSTM-GRU	Litecoin Monero	Technical-based	1,3,7 days	Record the lower value in the LSTM&GRU model compared with LSTM in metrics of MSE, RMSE, MAE, MAPE
Petrovic et al.	Adaptive neuro fuzzy inference system (ANFIS) architecture & Cauchy exploration strategy beetle antennae search (CESBAS) hybrid algorithm (CESBAS-ANFIS) LSTM LSTM-GRU	Litecoin Monero	Technical-based	1, 3, 7 days	Record the lower value in CESBAS-ANFIS model compared with LSTM and LSTM-GRU in metrics of MSE, RMSE, MAE, MAPE
Chowdhury et al.	Gradient Boosted Tree Neutral Network Ensemble learning method KNN	Bitcoin Dash DOGE Etherium IOIA Litecoin NEM NEO	Technical-based	week	92.4% accuracy using ensemble learning method, 90% accuracy using gradient boosted trees

Hitam, N. A., Ismail, A. R., and Saeed, F.	SVM-PSO	Bitcoin Ethereum Litecoin NEM Ripple Stellar	Technical-based	Day	90.4% accurancy for bitcoin, 97% for Ethereum, 92.1% for Litecoin, 57.8% for NEM, 82.8% for Ripple, 64.5% for Stellar
Sun et al.	Gradient Boosting Decision Tree (GBDT) Light Gradient Boosting Machine (LightGBM),	42 type crypto	Asset-based	Day Month Week	When the test set belongs to the training set or is independent of the training set, LightGBM, SVM, and RF models are better in 2 weeks than 2 days and 2 months predictive interval.
Felizardo et al.	ARIMA Random Forest (RF) Support Vector Machine (SVM) Long Short-Term Memory (LSTM) WaveNets	Bitcoin	Technical-based	1, 5, 10, 30 days	For prediction interval, which is 1 day and 5 days, SVM is the best performing model in MAE, MSE, RMSE, MAPE, and MPE evaluation metrics. For 10 days prediction interval, ARIMA and SVR is the best model used. Finally, for the 30days prediction interval, WaveNet or the LSTM performs better than other models.
Rathan et al.	Decision tree Linear regression	Bitcoin	Technical based	Day	Decision Tree with accuracy 95.88013 and linear regression with accuracy 97.59812
Derbentsev et al.	Random Forests (RF) Gradient Boosting Machine (GBM)	Bitcoin(BTC) Ethereum (ETH) Ripple (XRP)	Technical based	Day	Regarding MAPE, GBM has the highest prediction performance for Ripple, which is recorded as 0.92 percent, while RF produces the best outcome for Ripple, which is recorded as 1.84 percent.
Mohapatra et al.	KryptoOracle (Twitter sentiments) XGBoost	Bitcoin	Sentiment based& Technical based	Day	Between the actual and forecast Bitcoin prices, the accuracy is recorded as 10 USD as root mean square (RMS) error.

Borges et al.	Logistic Regression Random Forest Support Vector Classifier Gradient Tree Boosting Ensemble Voting	More than 100 crypto	Technical-based	Day Minute	Ensemble Voting, which combines the other four methods, has the highest average accuracy of 55.61 percent for all resampled market data.
Zhang et al.	ARIMA Support Vector Regression Random Forest Regressor XGBoost Regressor (XGB-Regressor) CNN LSTM Weighted Memory Channels Regression (WMCR)	Bitcoin Etherium Bitcoin cash	Technical-based	Day	WAMC has 77.52 accuracy at prediction of Ethereum market, and 75.23 for the Bitcoin Cash market
Kim et al.	ANN Support-vector machine (SVM)	Etherium Bitcoin Litecoin Dash	Blockchain-based Technical-based	Day	ANN records higher accuracy than SVM in all the models. Models I-4 (RMSE=0.068) and II-4 (RMSE=0.068) use ANN to record the highest accuracy with the lowest RMSE value.
Akyildirim et al.	Logistic regression SVM RF ANN	12main crypto	Technical-based	Day Minute Hour	SVM achieves the greatest accuracy compared to the Logistic regression, RF and ANN model. The average of all models is above 50 percent accuracy for all time stamps.
Peng et al.	Support Vector Regression (SVR)	Bitcoin Dash Etherium	Technical-based	Day	Compared to the nine GARCH models, SVR models had lower values when measured with RMSE and MAE error rates.
Kraaijeveld et al.	Vendor algorithm (Sentiment analysis)	Bitcoin Etherium	Technical-based	50 Day	All nine cryptocurrencies' polarity scores have mainly remained steady over time. With a mean polarity of 0.33, the scores are also

		Ripple Bitcoin Cash EOS Litecoin Cardano Stellar Tron			consistently positive biassed. Predictive power for price return is observed when to predict EOS and TRON when evaluation metrics bullishness ratio is used.
Gidea et al.	Topological analysis (K-means)	Bitcoin Etherium Litecoin Ripple	Technical-based	Day, hour	In total datasets, 6 out of 8 datasets record as true positive, which the other two records as false positive.
Phaladisailoed et al.	Theil-Sen Regression Huber Regression Long short-term memory (LSTM) Gated Recurrent Unit (GRU) Hibrid of NN and regression	Bitcoin	Technical-based	Day	GRU is recorded better than the other three methods with the lowest Mean Squared Error (MSE) value, which is 0.00002 and the highest R2 value, 99.2%.
Yogeshwaran et al.	Support vector machine (SVM) Convolutional Neural Networks (CNN) Recurrent neural networks (RNN) Long Short Term Memory (LSTM)	Bitcoin	Technical-based	Day	The four-layer method architecture is similar to that of a CNN. Among the three models, the three-layer technique performs the best. Following that is a four-layer technique, followed by a two-layer approach.
Samaddar et al.	Artificial neural network (ANN) Recurrent neural network (RNN)	Bitcoin price	Technical-based	Day	CNN has shown the best accuracy with 99.7%, as not only achieve higher accuracy also remain the least lost with 0.000162046 compared with other algorithms.

	Convolutional neural network (CNN) Random Forest (RF) k-nearest neighbors (k-NN)				
Yiying et al.	Artificial Neural Network (ANN) Long Short-Term Memory (LSTM) NN	Bitcoin Ethereum Ripple.	Technical-based.	7, 14, 21, 30, and 60 day	Using the ANN method, Bitcoin and Ripple show good prediction when the time interval is one day, while Ethereum shows good prediction when the prediction period is 3 days. Using LSTM as a predictive method, Ethereum and Ripple show good prediction when the time interval is 7 days, while Bitcoin shows good prediction when the prediction period is 14 days.
Politis et al.	LSTM GRU TCN Hybrid LSTM-GRU Hybrid LSTM-TCN Hybrid GRU-TCN Ensemble	Etherium	Technical-based.	1, 7days	The ensemble model reaches an accuracy of 84.2%.
Y. Indulkar	Long Short Term Memory (LTSM) Fbprophet	Bitcoin Ethereum Chainlink Bitcoin Cash Ripple	Technical-based.	1, 7, 15days, month, year	Bitcoin achieves the highest accuracy with the lowest MAE value, 0.04 for validation score and 0.02 for training score.
Waheeb et al.	Average Naive Drift	Bitcoin	Technical-based Blockcain-based.	1, 2, 3, 4, 5, 6 years	ELM achieves the highest accuracy when the validation set is used, with the lowest sMAPE score of 1.561081 and the highest sMAPE score of 8.678535 out of sample data.

	Auto-regressive integrated moving-average method (ARIMA) Simple exponential smoothing (SES) Holt exponential smoothing (Holt) Damped exponential smoothing (Damped) Combination method (COMB) Exponential smoothing (ETS) Bagged ETS (BaggedETS) Theta Multilayer perceptron method (MLP) Extreme learning machines (ELM)				
Akyildirim et al.	k-Nearest Neighbours Logistic regression Naive Bayes Random forest Support vector machine Extreme gradient boosting	Bitcoin	Technical-based.	5-, 10-, 15-, 30-, 60-min	At changing frequencies, the k-nearest neighbour (kNN) approach and the random forest (RF) algorithm produce the highest in- and out-of-sample accuracy rates. For example, the in-sample success rate for the random forest method can reach up to 87 percent for the first hold-out (0.7/0.3), and 83 percent for the second hold-out (0.8/0.3).
Shahbazi et al.	Reinforcement learning	Litecoin and Monero	Technical-based Blockcain-based.	3, 7, 30 days	The proposed method greatly enchance the RMSE value of the Litecoin and Monero in 3,7 and 30days price prediction.The greatest enchancement is form 14.0572 to 3.3097 for Litecoin and 16.1076 to 4.3826 for Monero.

Samaddar et al.	ANN CNN RNN Random Forent K-Nearest neighbour Neural Network	Bitcoin	Technical-based	minute	CNN is about 99.7%, which implies it had a very good prediction and only lost 0.000162046. RF has an accuracy of 99.9957.Though less accurate than both CNN and RNN, ANN also predicted reasonably well, with a loss of approximately 0.0740, which is greater than the loss for both CNN and RNN. The k-NN model predicted the data accurately as well, with a precision of 100.02331 percent, but with a loss of 0.6374, similar to the Random Forest model.
Freeda et al.	LSTM KNN GU SVM RF	Bitcoin Price	Technical-based	24-hour interval	Using LSTM an accuracy of 78.69% was achieved with a log loss of 7.18 to predict the direction of the close price. It is realized that recurrent neural model had lower RMSE and R squared value and also gave the minimum fluctuation in the bitcoin price
Tiwari et al.	ARIMA, Facebook Prophet XGBoost	Bitcoin	Technical-based	24-hour interval	The ARIMA, FB Prophet, and XGBoost have a root mean square error of 1165.14, 1300.81, and 13356.43, respectively. ARIMA is the most accurate machine learning technique for forecasting the bitcoin price based on the previous day's pricing.
E. Jakubowicz and E. Abdelfattah	Logistic Regression Support Vector Machine (SVM) Random Forest (RF) KNN Decision Tree (DT)	Bitcoin	Technical-based	minute	The Logistic Regression technique was extremely efficient, with an overall accuracy of 97 percent.The SVM model produced comparable results. Overall, it was 96 percent accurate.The Random Forest and Decision Tree models performed roughly identically, with the DT model outperforming the RF and all others. The accuracy values were

					99.9 percent and one hundred percent, respectively. The K Nearest Neighbor (KNN) model was the sole outlier. Its total accuracy was only 85%, and its F1 scores barely exceeded 90% on three of the eight levels.
Rane et al.	ARIMA Regression Model Latent Source Model (LSM) Binomial Generalized Linear Model (BGLM) Generalized Autoregressive Conditional Heteroskedasticity Model Support Vector Machine Model Long Short-Term Memory Network Model Non-linear Auto-Regressive with Exogenous Input Model Multi-Layer Perceptron Model	Bitcoin	Technical-based	Daily	Non-linear Auto-Regressive with Exogenous Input Model has the best accuracy at 62%. Among the other models, the Regression model (56%) has the best accuracy. Then the Multilayer perpeptron neural network model (55%), the Support Vector Machine model (54%), the ARIMA model (53%) and the LSTM model (52%) came in second and third, respectively (50 %).

2.5 Critical Summary

Overall, machine learning-based cryptocurrency pricing research is still in its early stages. This could be due to the protocol's novelty (Nakamoto, 2008), and machine learning techniques necessitate a large amount of data to understand connections between characteristics and target variables. The reviewed work has an explicit limitation: none of the papers has been published in a top-ranked finance or information systems journal. Furthermore, since we do the literature research based on the latest papers, the citation of the article review is recorded in the low number citation rate, which is low than 10 for most of the review articles, which do not have high representative power. Moreover, machine learning and cryptocurrency prediction are emerging disciplines. Therefore, our work reflects a short period of the literature in this field, and future analysis may produce different results.

Machine learning models are constructed and assessed on short periods and small data samples throughout the literature. Longer prediction intervals (e.g. weekly intervals) combined with powerful machine learning models and many characteristics may result in a sample with inadequate data points (Arnott et al., 2019). Furthermore, test splits of 3% or less, equivalent to 60 or fewer observations, limit the generalizability of the provided results (Atsalakis et al., 2019, Karakoyun et al., 2018). Furthermore, the difference in performance evaluation metrics due to the different methods, approaches, and datasets used causes the distinction of the outcome that is unfavourable to the comparative study between the papers. On the other hand, this study has the limitation of the typical comparative study, which cannot answer if more than one explaining variable occurs. Rather than that, there are more than 10000 cryptocurrencies in existence as of February 2022, based on the report of CoinGecko. Many of the cryptocurrencies have little and no trading volume. Nevertheless, some of the such as gaming based-cryptocurrency such as Smooth Love Potion (SLP), are widely used for the in-game digital currency of the Axie Infinity gaming metaverse, Sandbox (SAND) based in-game digital currency in the Sandbox metaverse and Decentraland (MANA) cryptocurrency to purchase virtual goods and services used in Decentraland platform. Gaming-based-cryptocurrency is the new use case of the cryptocurrency emerging after the term "Metaverse" that Facebook introduced. The recent study about machine learning to predict the currency's trend is limited to the mainstream currency. There is no current work and study about the use case of machine learning to forecast the gaming-based cryptocurrency. We encourage future researchers to evaluate machine learning for the gaming-based and the new-launched cryptocurrency such as Shiba Inu instead of predicting the mainstream.

2.6 Conclusion

Cryptocurrency has attracted great attention from scholars and investors since its inception in 2008. The research on cryptocurrency pricing via machine learning constitutes a relevant and emerging topic. A literature review of the existing body of research on cryptocurrency pricing via machine learning is conducted using Webster and Watson's and von Brooke et al. standards. The review is organized and analysed around four major concepts: method, feature, prediction interval, and prediction type. Most academics incorporate technical, blockchain-based, sentiment and interest-based, and asset-based considerations. However, a lack of transparency and comparability is noted throughout the literature, which limits the ability to validate and reproduce model results and trading approaches. It is advised that future researchers expose critical model configurations in a more organized manner, publish and upload their models and data to an open research repository, and benchmark their models against other published models.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, an overview of the research methodologies used throughout the study, from data collecting to model evaluation, is provided. The study aims to thoroughly examine the many available systems for forecasting metaverse-based cryptocurrencies. The chapter is divided into six sections. The section 1 provides an overview of the chapter. Section 2 provides the hardware specification and implementation environment. The data collection procedure is detailed in Section 3. The characteristics of the dataset and the data-preprocessing scheme are discussed in section 4. The proposed machine learning models that are implemented in this report are described in section 5. The model training and development details are provided in section 6. The model evaluation schemes are described in section 6. The overall of activities are detailed in section 8.

3.2 Implementation Environment

Table 2: Type of Hardware Requirement

No 1	Hardware Type	Specification
1	Processor	Intel(R) Core(TM)I7-7700HQ CPU @2.80GHz
2	RAM	8 GB
3	Storage	512 GB
4	Operating System	64-bit Window 10
5	Software and Tools	Google Colab

Table 2 shows the type of hardware requirement utilised in this project. The Google Colab is connected to the local run time and will be run using the Python 3.10.1 environment.

3.3 Data Collection

Data such as the trading volume and the historical price of Metaverse-based cryptocurrencies will be gathered from Yahoo finance's financial analysis website. The data set that tracks a historical daily closing price of meta cryptocurrencies over time will be obtained through these financial analysis websites by downloading the CSV files. The data features financial analysis websites' information such as date, open price, intraday highs and lows, closing price, and volume. Since Metaverse-based cryptocurrencies are the collection of observations of well-defined data items (closing price) obtained through repeated measurements over time, they can be defined as time-series data. Smooth Love Potion (SLP), The Sandbox (SAND), and Decentraland (MANA) are the Metaverse-based cryptocurrencies that will be selected as the predictive coins with these are the top metaverse coins by market capitalization.

The SLP coin, formerly known as the 'small love potion,' is a utility token that may be earned as a reward for playtime in the Pokémon-inspired Axie Infinity universe. The Ethereum blockchain underpins Axie Infinity. The SLP coin is an ERC-20 token. SLP can be used to breed digital pets known as Axies, which are NFTs that can be sold to other players on the Axie Infinity Marketplace.

SAND is the native cryptocurrency of The Sandbox, a popular 3D "metaverse" game built on the Ethereum network in which users can explore, buy land, and build monetizable constructions. All items purchased in The Sandbox are completely owned by the players, allowing them to profit from their purchases.

Decentraland (MANA) is a decentralised 3D virtual reality platform based on the Ethereum blockchain that allows users to develop virtual structures such as casinos, art galleries, music halls, and theme parks and charge other players to visit them. LAND, a non-fungible digital asset (ERC-721) divided into 16m x 16m chunks, is the accessible virtual environment within Decentraland. Community members hold these parcels in perpetuity, and they are acquired with MANA, Decentraland's native digital token. Some parcels are divided into themed communities known as Districts, which allow users to establish shared areas around shared interests.

The dataset of metaverse-based cryptocurrencies selected has six features shown in Table 3 below.

Table 3: Feature of the Metaverse-based Cryptocurrencies

Feature	Remarks
Date	The day on which an order to purchase, sell, or otherwise acquire a currency is completed in the market.
Open	The first price at which a currency is traded on a certain trading day
High	The highest price at which a currency is traded on a certain trading day.
Low	The lowest price at which a currency is traded on a certain trading day.
Close	The final price at which a currency is traded on a certain trading day.
Adj Close (Adjusted closing price)	The closing price after adjustments for all applicable splits and dividend distributions.
Volume	The totals quantity of contracts traded for a specified currency on a certain trading day.

Daily data of the predictive metaverse cryptocurrencies are provided. Individual datasets are described in depth below.

• Smooth Love Potion (SLP): July 8 ,2020 - April 18, 2022 (649 data points)

• The Sandbox (SAND): August 14, 2020 - April 18, 2022 (612 data points)

• Decentraland (MANA): November 9, 2017 - April 18, 2022 (1622 data points)

3.4 Data Pre-processing and Feature Engineering

Exploratory data analysis (EDA) must be implemented first to determine what kind of data is obtained and determine outliners detection. In this study, the other feature that excludes the closing price of metaverse-based cryptocurrencies will be the feature variable, and the target variable will be the closing prices of these cryptocurrencies. We will curate the new dataset by adding the additional features that we considered significant based on our domain knowledge and their significance in previous research in the cryptocurrency prediction field.

The preferred language used is Python, which contains numerous libraries widely used in machine learning. Pandas and NumPy library will be the tools to handle the dataset chosen in the data cleaning task. Pandas is well suited for many kinds of data such as SQL tables or Excel spreadsheets, order and unorder time series data, and arbitrary matrix data to handle the missing data, size mutability, and automatic and explicit data alignment. Numpy is a Python library that provides a simple yet powerful data structure, n-dimensional arrays to further operation on the arrays such as mathematical, logical, shape manipulation, sorting, and much more related to the data conversion. The platform used is the Google Colab, a free and open-source web tool that enables users to create and share documents that include live code, equations, visualisations, and narrative text. Panda and NumPy library is mainly used as the data cleaning process to remove the duplicate's value, maintain the correctness of the data, deal with the missing data, and data conversion. Sklearn is mainly used to normalise the data, making features more suitable for training by rescaling.

We must first convert the timestamp column to date because its data type is an integer rather than a date. We can use it to replace missing values to accommodate missing ones if they are frequently found in the dataset. Removal of the column with the null values can be used when the frequency of the null values existing in the dataset is low.

3.5 Model Choosing

The model is divided into two portions before picking an algorithm: 'Training data' and 'Validation data.' The datasets were divided into training and validation with an 80-20 ratio, with 80 percent used for training and the remaining 20% used for validation purposes. Moreover, the last ten rows of the dataset will be used as a testing set. We train the models during the training and validation period and then evaluate them during testing. The algorithms that are preferred to implement in this project are Long short-term memories (LSTM), Convolutional neural network (CNN), and Gated Recurrent Units (GRUs).

3.5.1 Long Short-Term Memory

Long Short-Term Memory (LSTM) is an RNN version capable of learning long-term dependencies. Although the construction of LSTMs is similar to that of RNNs, the repeating unit has a significantly different structure, as demonstrated in the Figure 5 below. They feature four neural network layers that interact with each other rather than just one.

An input gate, a forget gate, and an output gate make up a standard LSTM unit. These gates have the mathematical form indicated in the Equations below. These gates control the flow of information. The following are the structure of the LSTM cell.

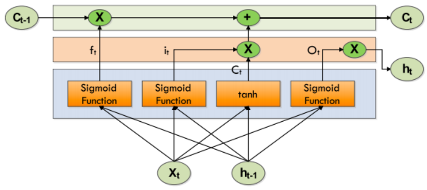


Figure 5: LSTM Cell Structure (Patel et al.)

The following equations summarize a LSTM.

$$i_t = \sigma(x_t V_i + h_{t\text{-}1} W_i)$$

$$f_t = \sigma(x_t V_i + h_{t\text{-}1} W_i)$$

$$o_t = \sigma(x_t V_o + h_{t-1} W_o)$$

$$\tilde{C} = tanh(x_t V_g + h_{t-1} W_g)$$

$$C_t = \sigma(f_t * C_{t\text{-}1} + \mathrm{i}_t * \tilde{C}_t)$$

where Forget Gate "f" (a neural network with sigmoid), Candidate layer "C`" (a neural network with Tanh), Input Gate "I" (a NN with sigmoid), Output Gate "O" (a NN with sigmoid), Hidden state "H" (a vector), Memory state "C" (a vectoR), x_t is input, h_{t-1} is previous cell output, C_{t-1} is previous cell memory, h_t is current cell output, C_t is current cell memory, and W, V denotes the weights.

3.5.2 Gated Recurrent Unit

Another RNN variation that solves the vanishing gradient problem is GRU. As introduced by (Cho et al., 2014), a GRU is comparable to an LSTM but contains fewer gates, as seen in Figure 6. It consists of two gates: an update gate and a reset gate. These two gates operate together to govern data flow through the network. The update gate determines how much information from the past needs to be sent to the next step. The reset gate determines the amount of data being forgotten. The formula used by the Gated Recurrent Unit is shown below:

$$u_t = \sigma\ (V_u x_t + W_u o_{t-1} + b_u)$$

$$r_t = \sigma\ (V_r x_t + W_r o_{t-1} + b_r)$$

$$i_t = tanh\ (V_o x_t + W_o (r_t \odot o_t - 1) + b_o)$$

$$o_t = u_t \odot o_{t-1} + (1 - u_t) \odot i_t$$

Where x_t is the input, ot is the output, ut is the update gate output, r_t is the reset gate output, $\odot$denotes the Hadamard product, and V, W, and b are the parameters or weight matrices.

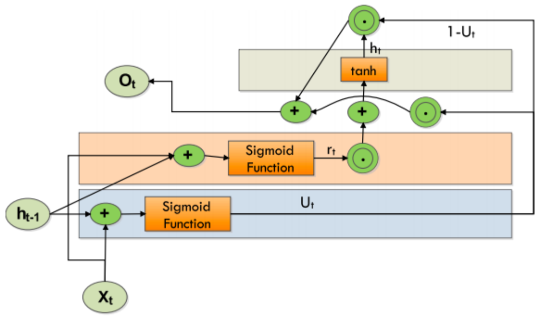


Figure 6: Structure of a GRU (Patel et al.)

3.5.3 Convolutional Neural Networks

CNN(convolutional neural network) are a type of neural network classification that has demonstrated high performance in fields such as image recognition. CNN is a neural feed forwarding network with multiple layers (Sharma et al., 2018).

CNN comprises neurons or filters with weights, parameters, and biases that can be adjusted (Lu et al., 2021). Convolutional, pooling, and fully connected layers make up CNN's structure. The feature extraction part of the CNN is where the input from each neuron is associated with the previous layer's local receptive field; the feature mapping part is where the input from each neuron is associated with the previous layer's local receptive field. In contrast to traditional recognition algorithms with complex extraction processes (Chen et al., 2014), CNN integrates extraction and classification in a single stage. Figure 7 illustrates the overall architecture of CNN.

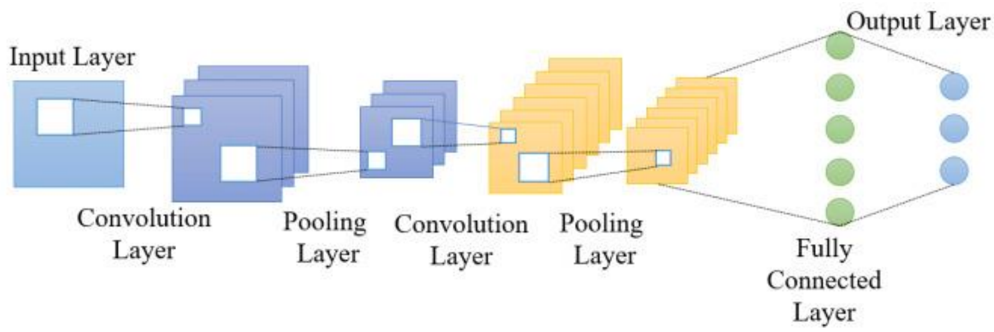


Figure 7: Architecture of the CNN

Convolution Layer

The convolutional layer is the central part of a Convolutional network that performs the most computational elevation. The goal of the convolution layer is to extract key features from image input data (Zhu et al., 2017). Convolution preserves the spatial relationship between pixels by learning image properties with tiny input squares. To convert the image into input, a variety of learning neurons can be used. This produces an activation map or map on the output image, which is then fed as input data into the next convolution layer, as shown in Figure 8 (Zhang et al., 2017).

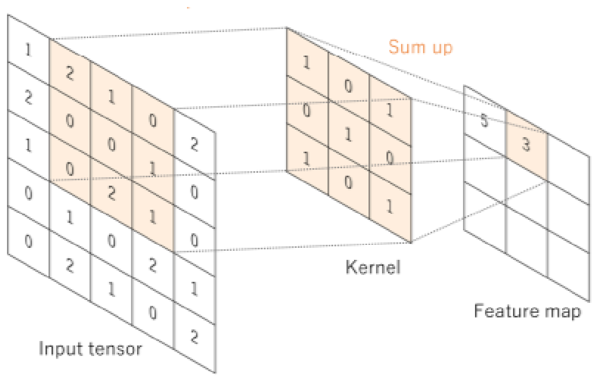


Figure 8: Convolution-Layer

Pooling Layer

The pooling layer, as represented in Figure 9, decreases the dimension of each activation map while keeping the essential information. The input photos are divided into rectangles that do not overlap. Each area will be sampled by a nonlinear activity such as limit or average (Albawi et al., 2017). This layer achieves more generalisation and faster integration and is more resistant to translation and distortion.

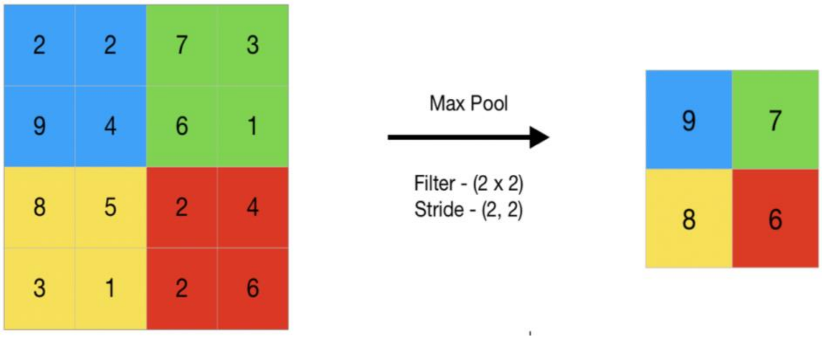


Figure 9: Pooling Layer

Fully Connected-Layer

The last few years of the network are made up of a fully connected layer (shown in Figure 10), which is a feed-forward neural network. The entrance to a fully connected layer is the output of the final pooling or convolution layer, which is flattened and then entered into a fully connected layer (Li & Zhang, 2020).

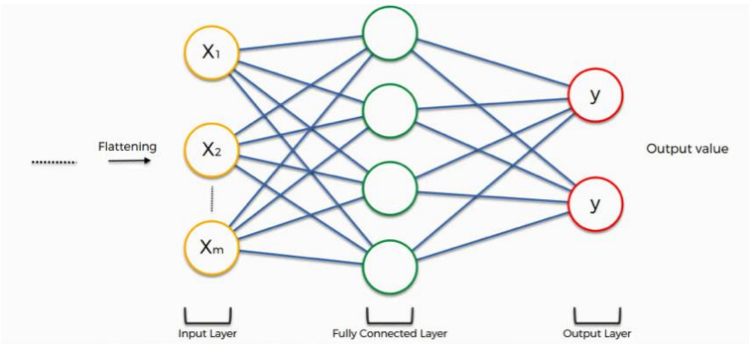


Figure 10: Fully Connected Layer

3.6 Model Training and Development

Keras is a high-level neural network library that works on top of TensorFlow. TensorFlow is an open-source end-to-end platform and library for different machine learning tasks. Keras and TensorFlow will be used to define the neural network model, such as determining the activation function and the number of nodes used as input variables for LSTM, CNN, and GRUs architecture for all metaverse-based cryptocurrencies datasets. The model after that is complied with and trained by using Keras with different optimisers, learning rates, epochs and batches. The hyperparameter tuning process runs experiments with different optimisers, learning rates, epochs, and batches to determine optimised models with the highest performance and accuracy.

3.7 Model Evaluation

For regression method evaluation metrics, RMSE (Root-mean-square deviation), MAE (mean absolute error), MSE (mean square Error), and MAPE (mean absolute percentage error) are used to evaluate the efficiency and performance of the machine learning model toward the metaverse-based cryptocurrencies price.

The MAE represents the average of the absolute difference between the actual and predicted values in the dataset. MSE represents the average of the squared difference between the original and predicted values in the data set. RMSE is the square root of the Mean Squared error. MAE measures the average of the residuals in the dataset, whereas MSE and RMSE measure the residuals' variance and the standard deviation of residuals. Mean absolute percentage error (MAPE) means or the average of the absolute percentage errors of forecasts. The lower value of MAE, MSE, RMSE, and MAPE implies higher accuracy of a regression model. The different models are compared with the optimal tested hyperparameter to evaluate the best model suitable for the metaverse-cryptocurrencies prediction. The metaverse cryptocurrencies' MAE, MSE, RMSE, and MAPE values will be tabulated in the tables. In these metrics, MAPE will use as the primary metrics as the MAPE, where MAPE allows the error to be compared across data with different scales that are favoured to the comparison between the different metaverse cryptocurrencies.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - x|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - x)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - x)^2}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - x}{x}\right|$$

where X_i represents the predicted price, X represents the actual price and N is total number of observations.

3.8 Overall Flow of Activities

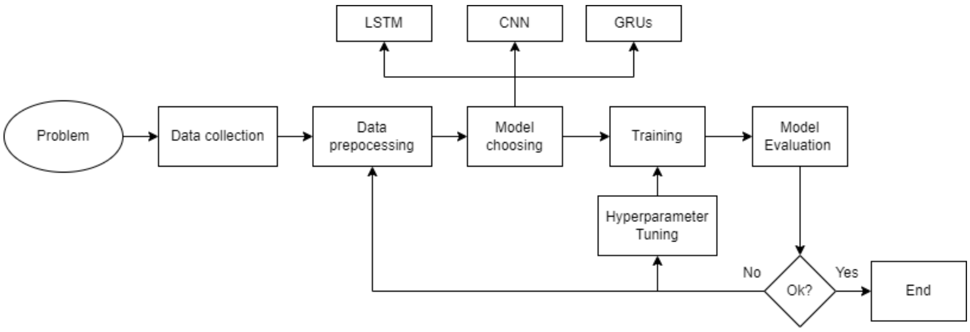


Figure 11: Overall Flow of Activities

The overall flow of activities for the proposed project is illustrated in Figure 11. The process begins with the identification of the problem statement, followed by data collection and pre-processing steps. The appropriate model for the problem is then chosen from among LSTM, CNN, and GRUs models. The chosen model is then developed and evaluated. Hyperparameter tuning is performed on the model, and it is retrained until satisfactory results are achieved.

3.9 Conclusion

In conclusion, chapter 3 describes the methodology's overview, implementation environment and hardware specification, type of dataset selected, and the feature of the data. The project methodology is explained explicitly throughout the chapter, including all stages of the machine learning framework for the metaverse cryptocurrencies implemented in the preliminary experiment.

CHAPTER 4

EXPERIMENTAL DESIGN

4.1 Introduction

This section introduces the experimental design of the three proposed models for the price prediction of three different metaverse cryptocurrencies, which are Smooth Love Potion (SLP), Sandbox (SAND) and Decentraland (MANA), by using different input features in the three different experiments. The proposed models are built separately by Convolutional Neural Networks architecture, Long Short-Term Memory and Gated recurrent unit and the architecture of the proposed models are described in this section.

RNNs are built in a way that will allow them effectively analyze sequence data. They have a type of memory called sequential memory. Their structure is such that their output is a combination of current inputs and information learnt from prior inputs and outputs. As a result, they are effective at sequence-related tasks such as time-series prediction. However, they suffer from the vanishing gradient problem, making them more difficult to train. Both LSTM and GRU are RNN variations meant to avoid the vanishing gradient problem that RNNs faces. Many prior studies have demonstrated the superiority of LSTM and GRU in time-series prediction in traditional cryptocurrencies. Convolutional Neural Network (CNN) models are mainly used for two-dimensional arrays like image data. Furthermore, in most of the sequences in traditional cryptocurrency prediction, CNN models are normally used for binary classification instead of regression to determine the buy-and-hold decision instead of predicting the movement and the price of the cryptocurrencies. However, CNN can be applied with regression data analysis for time series problems such as metaverse cryptocurrencies prediction with to output regression outputs instead of the class output.

Furthermore, data feature engineering is also discussed in this section about the algorithm involved in data and feature engineering for both SAND, SLP and MANA experiments. The parameter and architecture of the three proposed models are also discussed in this section. At the end of this section, the experiment set up for three experiments are described in diagram form.

4.2 Data and Feature Engineering

The first step is preparing the data and making it suitable for input to the three models. The datasets of three metaverse cryptocurrencies which are Sandbox with labelled SAND (shown in Figure 12 below), Smooth Love Potion with labelled SLP (shown in Figure 13 below), and Decentraland with labelled SAND (shown in Figure 14 below), are used are the classical financial dataset which consists of 5 features that are implicit discussed in Table 3, which are Open, High, Low, Close, Adjusted Close Price and Volume.

	Open	High	Low	Close	Adj Close	Volume
Date						
2020-08-14	0.086577	0.086577	0.059597	0.065645	0.065645	118988161
2020-08-15	0.066077	0.069509	0.048985	0.050964	0.050964	36213441
2020-08-16	0.051438	0.069508	0.048064	0.060847	0.060847	33023213
2020-08-17	0.060847	0.065073	0.052525	0.053999	0.053999	11686493
2020-08-18	0.053784	0.060990	0.050675	0.051700	0.051700	7401293
...	...	...	...	...	...	...
2022-04-13	2.781983	2.886945	2.751847	2.869923	2.869923	321784424
2022-04-14	2.869366	2.916186	2.714203	2.780882	2.780882	349909147
2022-04-15	2.781060	2.820187	2.755915	2.803639	2.803639	189256519
2022-04-16	2.803614	2.901280	2.777383	2.822680	2.822680	234480315
2022-04-17	2.820111	2.820111	2.781153	2.790278	2.790278	176793328

612 rows × 6 columns

Figure 12: SAND Dataset

	Open	High	Low	Close	Adj Close	Volume
Date						
2020-07-08	0.199600	0.200201	0.109705	0.110012	0.110012	8412
2020-07-09	0.110270	0.149273	0.062864	0.091263	0.091263	5864
2020-07-10	0.091247	0.104638	0.045408	0.055206	0.055206	302
2020-07-11	0.055206	0.089824	0.023832	0.033257	0.033257	380
2020-07-12	0.033253	0.033525	0.018149	0.018416	0.018416	42
...	...	...	...	...	...	...
2022-04-13	0.016956	0.017662	0.016659	0.017561	0.017561	195753852
2022-04-14	0.017563	0.018326	0.016753	0.017127	0.017127	214228125
2022-04-15	0.017128	0.017279	0.016853	0.017239	0.017239	107932100
2022-04-16	0.017242	0.017618	0.016952	0.017152	0.017152	128101400
2022-04-17	0.017156	0.017343	0.017092	0.017194	0.017194	110862176

649 rows × 6 columns

Figure 13: SLP Dataset

	Open	High	Low	Close	Adj Close	Volume
Date						
2017-11-09	0.014329	0.016111	0.013886	0.015130	0.015130	653800
2017-11-10	0.015162	0.015734	0.012645	0.012940	0.012940	296429
2017-11-11	0.013201	0.013616	0.011758	0.011979	0.011979	237865
2017-11-12	0.011975	0.011975	0.010125	0.010505	0.010505	242761
2017-11-13	0.010628	0.011660	0.010628	0.011330	0.011330	225836
...	...	...	...	...	...	...
2022-04-13	2.153283	2.212506	2.113025	2.201286	2.201286	216991854
2022-04-14	2.200854	2.235230	2.100408	2.117604	2.117604	269798320
2022-04-15	2.117844	2.152222	2.107485	2.143937	2.143937	201367474
2022-04-16	2.143889	2.184819	2.117178	2.152566	2.152566	159910112
2022-04-17	2.154083	2.154435	2.125707	2.133954	2.133954	125790432

1621 rows × 6 columns

Figure 14: MANA Dataset

The initial rows and columns for the SAND dataset, SLP dataset and MANA dataset are recorded as 612 rows, 649 rows and 1621 rows, and the columns for both three datasets are the same, which are six columns. The SAND dataset recorded the metaverse cryptocurrencies' time series data with five features discussed in Table 3 starting from the release date, which is 14 August 2020, until 17 April 2020. The SLP dataset recorded the metaverse cryptocurrencies' time series data with five features discussed in Table 4 starting from the release date, which is 8 July 2020, until 17 April 2020. The MANA dataset recorded the metaverse cryptocurrencies' time series data with five features discussed in Table 3 starting from the release, which is 8 November 2017, until 17 April 2020.

The rows in a time-series data type, called elements, each represent one or more data values for a specific time stamp. The column indicates the features of the metaverse cryptocurrencies data, which are Open, High, Low, Close, Adjusted Close Price and Volume. The column 'Adjusted Close Price' feature will be eliminated from the dataset as the data is the same for the closing price 'Close' since cryptocurrencies do not undergo stock splits dan dividend distributions. The previous closing price labelled 'Pre_Close' has been added to the dataset as the new feature. This feature indicates the last closing price of the datasets with a timestamp minus one.

We added five additional features that we considered significant based on our domain knowledge and their significance in previous research into the dataset. The dataset consisted of technical indicators commonly used in the stock and Foreign Exchange (Forex) market trade. The additional features are included the differences between the highest and the lowest price at which a currency is traded on a certain trading day (H-L), differences between the first price and the last price at which a currency is traded on a certain trading day (O-C), the Simple Moving Average of past seven days (SMA_7), the Exponential Moving Average of past seven days (EMA_7) and the Relative Strength Index of past 7 days (rsi_7).

The SMA_7 calculates the average of seven days of the closing price. It averages the closing prices for the first seven days as the first data point. The following data point would drop the earliest price, add the price on day 7, take the average, and so on. Likewise, a 7-day moving average would accumulate enough data to average seven consecutive days of data on a rolling basis. The formula for SMA is shown below:

$$SMA = \frac{A_1 + A_2 + A_n}{n}$$

Where A_n is the closing price of the metaverse cryptocurrencies at period n, n is the number of total periods.

The exponential moving average (EMA) is a technical chart indicator that follows the price of the stock or commodity over time. The EMA, as opposed to the simple moving average (SMA), is a weighted moving average (WMA) that lends greater weight or relevance to recent price data. The EMA is intended to improve on the concept of the SMA by assigning more weight to the most recent price data, which is seen to be more relevant than older data. Because new data bears more weight, the EMA reacts to price fluctuations faster than the SMA. The EMA_7 is a seven-day-moving average that places a greater weight and significance on the most recent data points. The formula for EMA is shown below:

$$EMA = Price(t) \times k + EMA(y) \times (1 - k)$$

Where t is today, y is yesterday, N is the number of days in EMA and k=2÷(N+1).

The relative strength index (RSI) is a momentum indicator used in technical analysis that examines the degree of recent closing price fluctuations to determine if a stock or other asset is overbought or oversold. It is common to be used in the stock and forex trade.

The average gain or loss used in the calculation is the average percentage gain or loss during a look-back period. The formula uses a positive value for the average loss. Periods with price losses are counted as 0 in the calculations of average gain, and periods when the price increases are counted as 0 for the analysis of average losses. The rsi_7 means the average percentage gain or loss during the past seven days. The formula for RSI is shown below:

$$RSI_{step1} = 100 - \left[\frac{100}{1 + \frac{Average\ gain}{Average\ loss}} \right]$$

$$RSI_{step2} = 100 - \left[\frac{100}{1 + \frac{(Previous\ Average\ Gain\ \times 6) + Current\ Gain}{(Previous\ Average\ Gain\ \times 6) + Current\ Loss}} \right]$$

The average gain or loss used in the calculation is the average percentage gain or loss during a look-back period. The formula uses a positive value for the average loss. Periods with price losses are counted as 0 in the calculations of average gain, and periods when the price increases are counted as 0. After the addition of 6 new features and the elimination of the "Adjusted Closing price, both three datasets have 11 columns which indicate 11 features. For example, the SAND datasets below have 612 rows and 11 columns. The Pre_Close column has a null value at the first row for both datasets because the Pre_Close is made up of the Close column by shifting down 1 column. The "rsi_7" column has six null values because the seven days relative strength index makes use of the recent six closing prices to predict the next days day's relative strength index. Since the null values columns are comparatively small in both three datasets, the ways used to handle missing values are by deleting the rows or columns having null values.

Date	Open	High	Low	Close	Volume	Pre_Close	H-L	0-C	SMA_7	rsi_7	EMA_7
2020-08-14	0.086577	0.086577	0.059597	0.065645	118988161	NaN	0.026980	0.020932	0.065645	NaN	0.065645
2020-08-15	0.066077	0.069509	0.048985	0.050964	36213441	0.065645	0.020524	0.015113	0.058304	NaN	0.057256
2020-08-16	0.051438	0.069508	0.048064	0.060847	33023213	0.050964	0.021444	-0.009409	0.059152	NaN	0.058809
2020-08-17	0.060847	0.065073	0.052525	0.053999	11686493	0.060847	0.012548	0.006848	0.057864	NaN	0.057050
2020-08-18	0.053784	0.060990	0.050675	0.051700	7401293	0.053999	0.010315	0.002084	0.056631	NaN	0.055296
...	...	...	...	...	...	...	...	...	...	...	...
2022-04-13	2.781983	2.886945	2.751847	2.869923	321784424	2.782421	0.135098	-0.087940	2.886363	40.236566	2.909637
2022-04-14	2.869366	2.916186	2.714203	2.780882	349909147	2.869923	0.201983	0.088484	2.843185	36.188051	2.877448
2022-04-15	2.781060	2.820187	2.755915	2.803639	189256519	2.780882	0.064272	-0.022579	2.829312	38.046757	2.858996
2022-04-16	2.803614	2.901280	2.777383	2.822680	234480315	2.803639	0.123897	-0.019066	2.803156	39.759602	2.849917
2022-04-17	2.820111	2.820111	2.781153	2.790278	176793328	2.822680	0.038958	0.029833	2.788510	37.690803	2.835007

612 rows × 11 columns

Figure 15: Null values Inside the SAND Dataset

After removing the missing values column, the current rows and columns for the SAND dataset, SLP dataset and MANA dataset are recorded as 606 rows, 643 rows and 1615 rows, and the columns for both datasets are the same, which are six columns. The current SAND dataset (shown in Figure 16) recorded the metaverse cryptocurrencies time series data with 11 features discussed in Table 4 starting from the release date, which is 20 August 2020, until 17 April 2020. The final SLP dataset (shown in Figure 17) recorded the data with five features discussed in Table 4starting from its release date, which is 14 July 2020, until 17 April 2020. The final MANA dataset (shown in Figure 18) recorded the data with 11 features discussed in table 4 starting from its release date, 15 November 2017, until 17 April 2020.

Date	Open	High	Low	Close	Volume	Pre_Close	H-L	0-C	SMA_7	rsi_7	EMA_7
2020-08-20	0.052095	0.057325	0.050800	0.053648	7332475	0.052095	0.006525	-0.001553	0.055557	36.094661	0.054128
2020-08-21	0.053644	0.053738	0.044235	0.044235	5121097	0.053648	0.009503	0.009409	0.052498	23.308807	0.051380
2020-08-22	0.044395	0.049011	0.040692	0.046011	4347394	0.044235	0.008319	-0.001616	0.051791	28.856159	0.049928
2020-08-23	0.045990	0.048237	0.041965	0.044007	3114426	0.046011	0.006272	0.001983	0.049385	26.347291	0.048360
2020-08-24	0.043902	0.053543	0.042178	0.051299	7868186	0.044007	0.011365	-0.007397	0.048999	46.203257	0.049127
...	...	...	...	...	...	...	...	...	...	...	...
2022-04-13	2.781983	2.886945	2.751847	2.869923	321784424	2.782421	0.135098	-0.087940	2.886363	40.236566	2.909637
2022-04-14	2.869366	2.916186	2.714203	2.780882	349909147	2.869923	0.201983	0.088484	2.843185	36.188051	2.877448
2022-04-15	2.781060	2.820187	2.755915	2.803639	189256519	2.780882	0.064272	-0.022579	2.829312	38.046757	2.858996
2022-04-16	2.803614	2.901280	2.777383	2.822680	234480315	2.803639	0.123897	-0.019066	2.803156	39.759602	2.849917
2022-04-17	2.820111	2.820111	2.781153	2.790278	176793328	2.822680	0.038958	0.029833	2.788510	37.690803	2.835007

606 rows × 11 columns

Figure 16: Final SAND Dataset

Date	Open	High	Low	Close	Volume	Pre_Close	H-L	0-C	SMA_7	rsi_7	EMA_7
2020-07-14	0.012000	0.076397	0.011938	0.034327	419	0.011982	0.064459	-0.022327	0.050638	27.681177	0.036470
2020-07-15	0.034318	0.034480	0.010670	0.010735	79	0.034327	0.023810	0.023583	0.036455	20.642657	0.029320
2020-07-16	0.010729	0.010755	0.009543	0.009627	35	0.010735	0.001212	0.001102	0.024793	20.359012	0.023997
2020-07-17	0.009628	0.009658	0.009562	0.009590	35	0.009627	0.000096	0.000038	0.018276	20.348119	0.020181
2020-07-18	0.009594	0.011825	0.009574	0.011774	42	0.009590	0.002251	-0.002180	0.015207	23.178636	0.017986
...	...	...	...	...	...	...	...	...	...	...	...
2022-04-13	0.016956	0.017662	0.016659	0.017561	195753852	0.016955	0.001003	-0.000605	0.018427	37.914164	0.018445
2022-04-14	0.017563	0.018326	0.016753	0.017127	214228125	0.017561	0.001573	0.000436	0.017850	35.801746	0.018116
2022-04-15	0.017128	0.017279	0.016853	0.017239	107932100	0.017127	0.000426	-0.000111	0.017545	36.860883	0.017897
2022-04-16	0.017242	0.017618	0.016952	0.017152	128101400	0.017239	0.000666	0.000090	0.017231	36.317886	0.017710
2022-04-17	0.017156	0.017343	0.017092	0.017194	110862176	0.017152	0.000251	-0.000038	0.017078	36.841893	0.017581

643 rows × 11 columns

Figure 17: Final SLP Dataset

Date	Open	High	Low	Close	Volume	Pre_Close	H-L	O-C	SMA_7	rsi_7	EMA_7
2017-11-15	0.011461	0.012014	0.011398	0.012002	283268	0.011419	0.000616	-0.000541	0.012186	33.965216	0.011807
2017-11-16	0.012013	0.012595	0.011742	0.011793	357512	0.012002	0.000853	0.000220	0.011710	31.878810	0.011803
2017-11-17	0.011729	0.012085	0.011187	0.012028	268762	0.011793	0.000898	-0.000299	0.011579	36.958740	0.011864
2017-11-18	0.012021	0.013352	0.011435	0.013019	370149	0.012028	0.001917	-0.000998	0.011728	53.879565	0.012170
2017-11-19	0.012934	0.013975	0.012661	0.013099	379424	0.013019	0.001314	-0.000165	0.012099	55.016698	0.012413
...	...	...	...	...	...	...	...	...	...	...	...
2022-04-13	2.153283	2.212506	2.113025	2.201286	216991854	2.153636	0.099481	-0.048003	2.243642	37.935015	2.251587
2022-04-14	2.200854	2.235230	2.100408	2.117604	269798320	2.201286	0.134822	0.083250	2.200956	33.194641	2.218091
2022-04-15	2.117844	2.152222	2.107485	2.143937	201367474	2.117604	0.044737	-0.026093	2.183354	36.124984	2.199553
2022-04-16	2.143889	2.184819	2.117178	2.152566	159910112	2.143937	0.067641	-0.008677	2.156682	37.178455	2.187806
2022-04-17	2.154083	2.154435	2.125707	2.133954	125790432	2.152566	0.028728	0.020129	2.136927	35.696953	2.174343

1615 rows × 11 columns

Figure 18: Final MANA dataset

After that, the continuous step is to prepare the data and make it suitable for input to the model. We will use the different dependent features to forecast the target variable through the experiments discussed in the later session. In all experiments that are done in the below sections, the dependent features vary when fitting to the model. The target features 'Close' is always the same to determine the predictive power of the different dependent features to predict the closing price. After the feature selection steps, the min-max normalization method is used for the data normalization to convert the values of the dependent features to the target features in the range of 0 to 1. The min-max normalization formula is shown below:

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

Where x is an original value, x' is the normalized value.

The dependent features used in the target features were bifurcated into training & validation with an 80-20 ratio; 80% was for the training purpose, and the remaining 20 % was for validation purposes based on the length of the dataset. The predicted horizon is also the same for three experiments, which are ten days. The look-back period for the three experiments is the same: ten days, meaning the last ten days' dependent features are used to predict the next ten days' closing prices. For example, after the data preprocessing step, the SAND datasets have the first 474 rows as training datasets, and the left 102 rows are for the validation datasets. For

SLP datasets, the first 504 rows are training datasets, and the remaining 109 are for the validation datasets. For MANA datasets, the first 504 rows as training datasets and the left 109 rows are for the validation datasets. The last ten days after the three data sets are not fed into the models as the usage of testing data and check the prediction against it by visualizing the actual and predicted values. The visualization for cryptocurrency bifurcation based on training & validation data can be seen in Figure 19, Figure 20 and Figure 21, shown below.

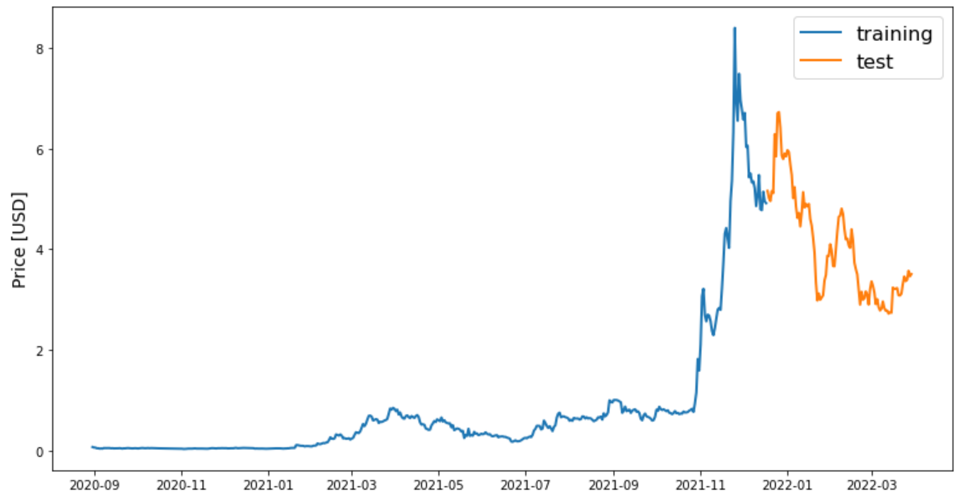


Figure 19: Dataset Bifurcation for SAND into Training and Validation

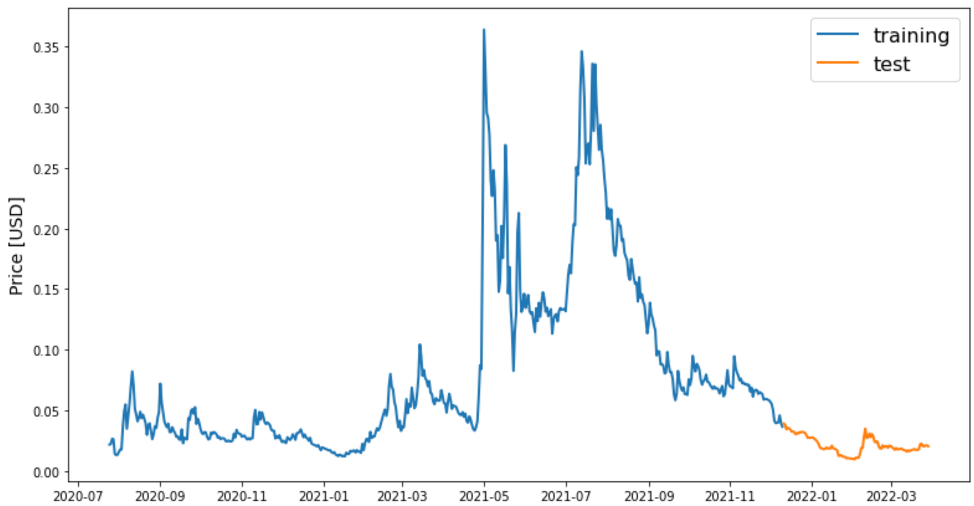


Figure 20: Dataset Bifurcation for SLP into Training and Validation

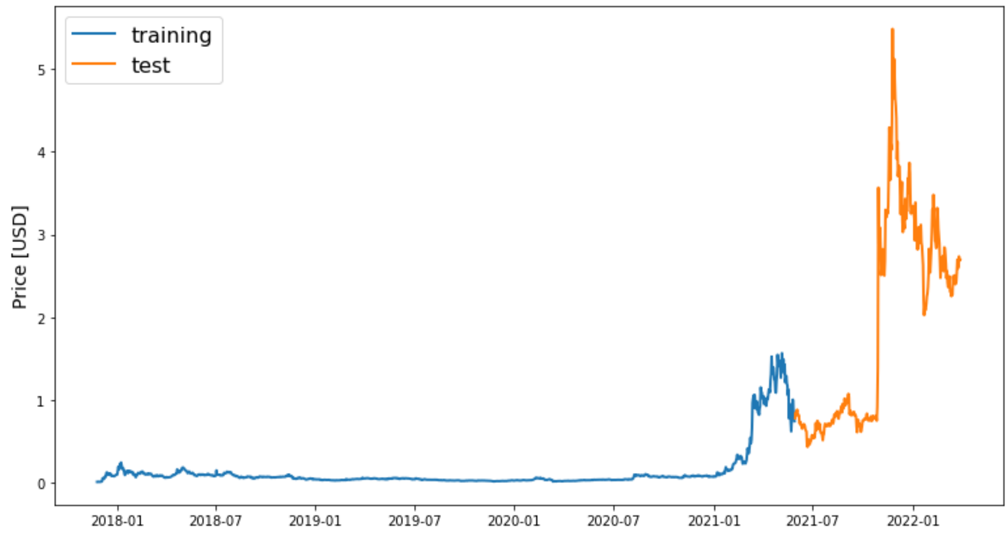


Figure 21: Dataset Bifurcation for MANA into Training and Validation

4.3 Model Design

After the data preprocessing, the dataset with the selected dependent feature and target feature can be fed separately to the three models discussed later. The algorithms used for the metaverse cryptocurrencies prediction are LSTM, GRUs, and CNN models. The sections below discuss the architecture of LSTM, GRUs, and CNN proposed models. The parameters used for these three models are standardized, as shown in the Table 4 below.

Table 4: Parameter Value of LSTM, GRUs and CNN Models

Parameter Value	Value
epochs	50
batch size	32
Loss function	Mean square error
optimizer	Adam

4.3.1 LSTM Model

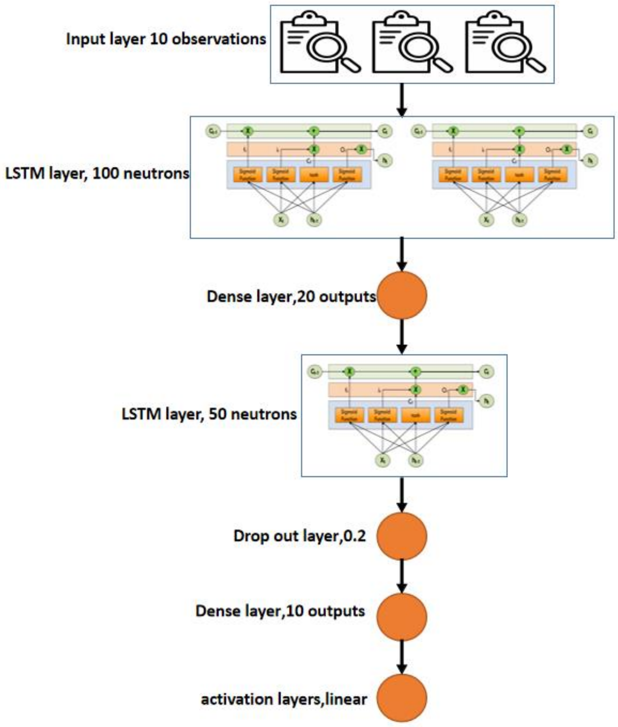


Figure 22: The Architecture of LSTM Model

The LSTM model, as shown in Figure 22, adapts the LSTM model developed by Y. Indulkar (2021). The researcher employs one layer of LSTM with 100 neurons, a dropout layer with a dropout value of 0.2, and a final dense layer with a Linear activation function to produce one output for the prediction value. Our model contains one input layer with 100 LSTM input nodes connected to a dense layer with 20 outputs and then to a dropout layer with a dropout value of 0.2 to prevent overfitting of the model, which is densely connected to the final layer with a linear activation function. The optimizer employed for the model was Adam, and the loss function was MSE (Mean Squared Error). The model was fit to the training data using 50 epochs and 32 batches, respectively. The Adam optimizer can be understood through the below equations.

$$\theta_{t-1} = \theta_t - \frac{\eta}{\sqrt{v_t + \epsilon}}\widehat{m_t}$$

$$\widehat{m_t} = \frac{m_t}{1 - \beta_2^t}$$

$$\widehat{V_t} = \frac{v_t}{1-\beta_2^t}$$

Where, the m_t & v_t are the values of vectors that are initialized to 0's. Further, the biases in the optimizer can be seen in respectively, η is the learning rate.

4.3.2 GRUs model

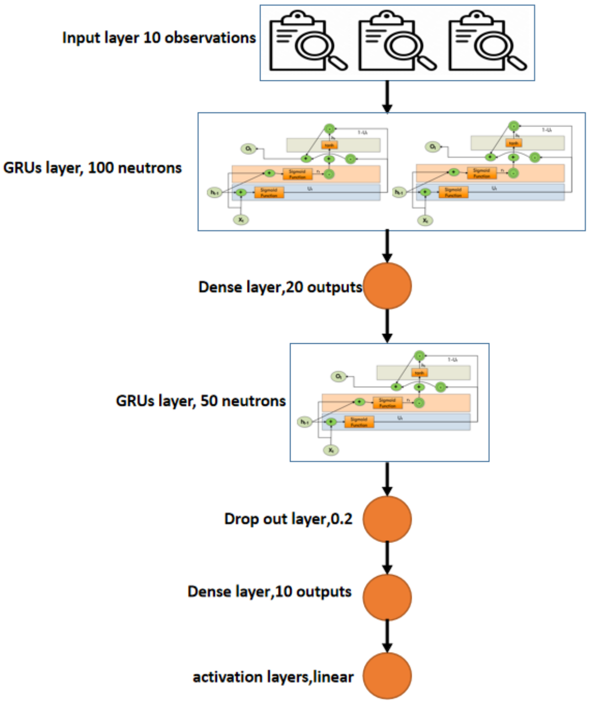


Figure 23: The Architecture of GRUs Model

Similarly, we construct the GRUs model shown in Figure 23 by changing the LSTM layers to GRUs layers. GRUs model contains one input layer with 100 GRUs input nodes connected to a dense layer with 20 outputs and then to a dropout layer with a dropout value of 0.2 to prevent overfitting of the model, which is densely connected to the final layer with a linear activation function. The optimizer employed for the model was Adam, and the loss function was MSE (Mean Squared Error). The model was fit to the training data using 50 epochs and 32 batches, respectively.

4.3.3 CNN model

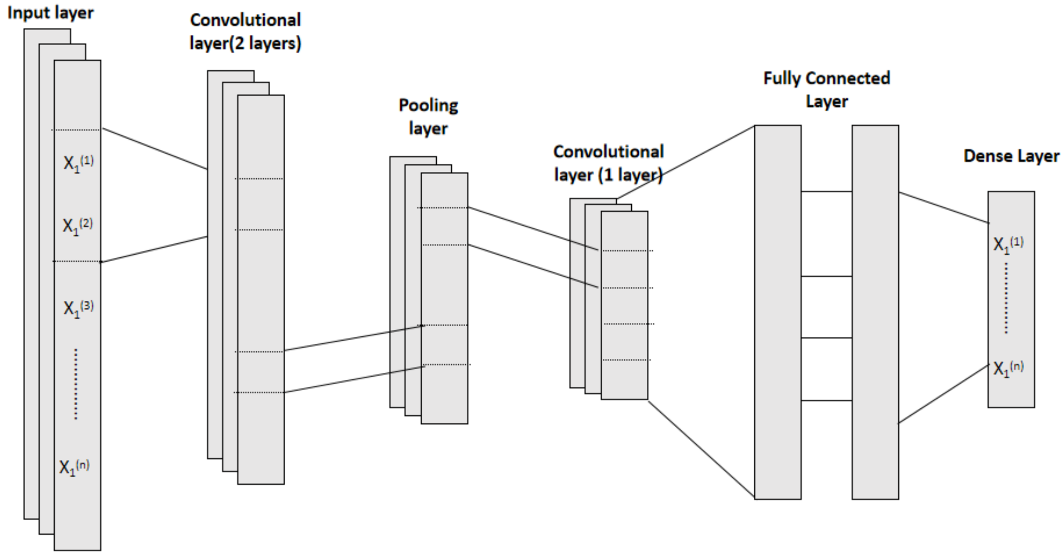


Figure 24: The Architecture of CNN Model

The CNN model, as shown in Figure 24, is an adaptation of the CNN model developed by Chen, S., & He, H. (2018). Since the data of our input dataset belongs to 1D time-series data, the convolutional function that has been used is "Conv1D". A large number of convolutional layers may result in complex computation and gradient vanishing or diffusion, whereas a limited number of convolutional layers may produce unreliable results. Therefore, three layers of convolutional layers are chosen to strike a balance between the speed of computing and the efficacy of our model. As shown in the architecture of the CNN model above, the input layers are connected to the two layers of convolutional layers. The convolutional layers are connected to the pooling layers to reduce the dimension of the activation map while keeping the essential information. After that, the pooling layers are connected to a convolutional layer. The size of the filter of Conv1D is 32, 64 and the first two convolutional layers and 128 for the others. The convolutional layer is flattened and connected to the fully connected layer. The fully connected layer consists of 1 dense layer with 512 neurons and a 0.2 drop-out layer. The fully connected layer is further connected with the dense layer with an output size of 10. The size of kernels in the pooling layer and both three convolutional layers is 1. The optimizer employed for the model was Adam, and the loss function was MSE (Mean Squared Error).

4.4 Model Evaluation

The loss curve is observed after feeding the dependent features into the three models separately. The testing data for the last ten days and check the prediction against it by visualizing the actual and predicted values. Finally, the result is evaluated with standard performance metrics. The overall system architecture is shown as Figure 25 as below:

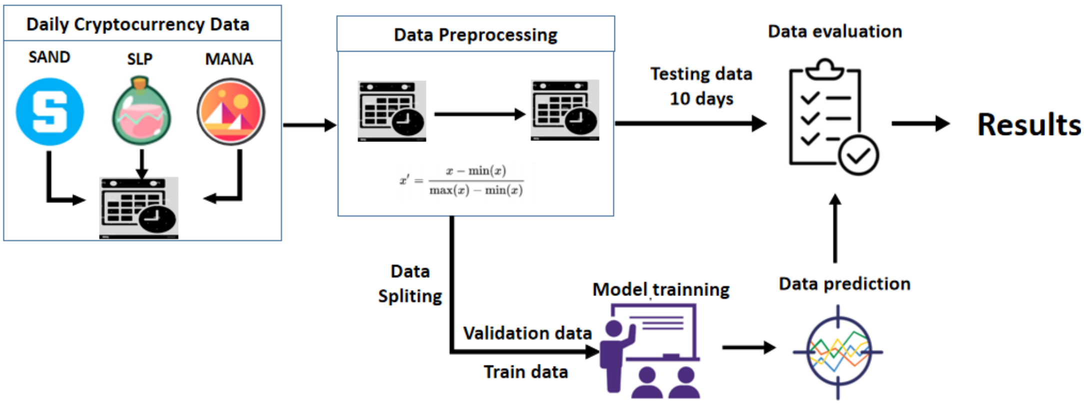


Figure 25: System Architecture

4.5 Feature Selection Experiments Setup

The experiments are setup up to compare the predictive power of the 11 dependent features, which are Open, High, Low, Close, Volume, Pre_Close, H-L, O-C, SMA_7, rsi_7 and EMA_7, towards the target and predictive variable. To check the predictive power of different features towards the target variables with three cryptocurrencies, SAND, SLP and MANA, the three experiments are set up with the various input variables.

4.5.1 Experiments 1 Setup

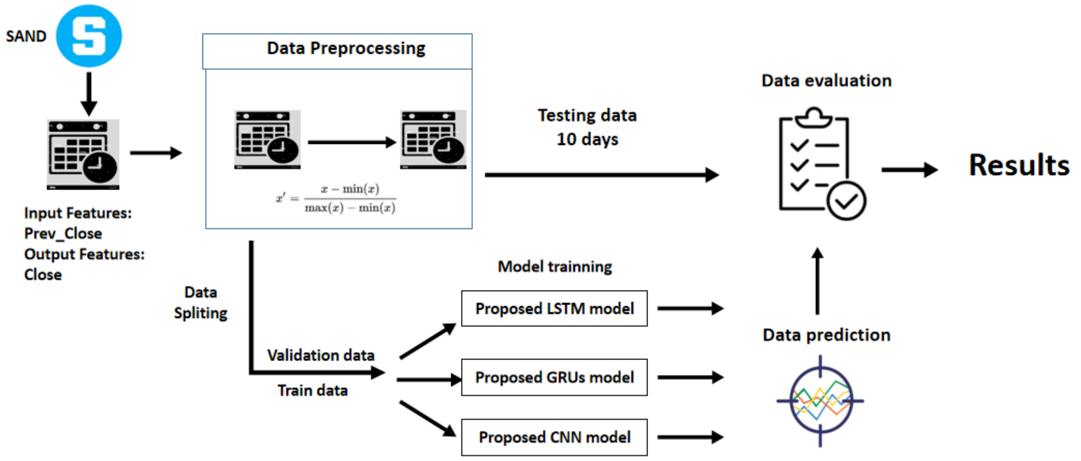


Figure 26: Experiments 1 Setup for SAND

Experiment 1 is set up like the Figure 26 for the SAND metaverse cryptocurrency. The input feature is Pre_Close which means the Previous Closing Price, and the output feature and variable is Close, which means the closing price. The importance of experiment 1 is to check the performance and efficiency of using previous closing price as an input variable to predict the next ten days' closing price of the metaverse cryptocurrencies. After processing the data preprocessing and feature selection, the data is split as 80-20 ratio, 80 % was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrency price and compared with the actual price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales. The same procedure will be repeated using SLP and MANA datasets shown in Figure 27 and Figure 28 below.

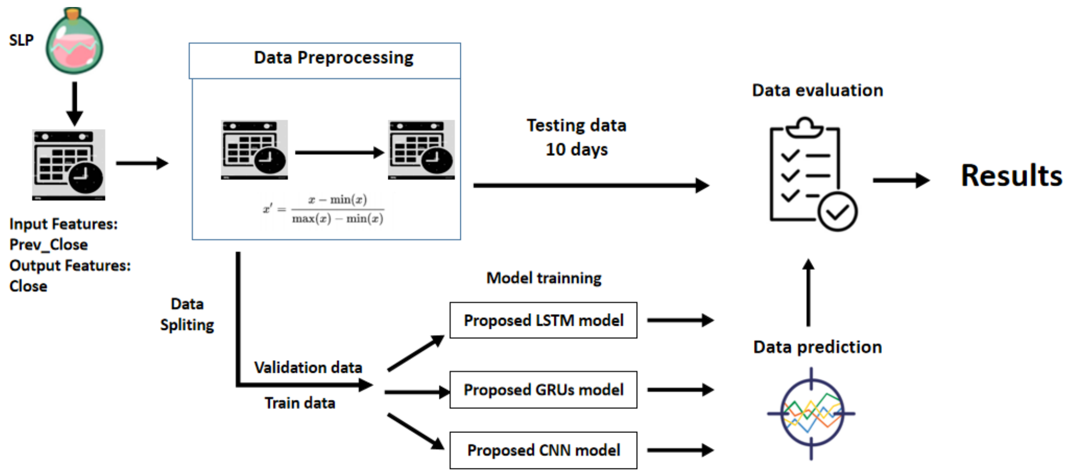


Figure 27: Experiments 1 Setup for SLP

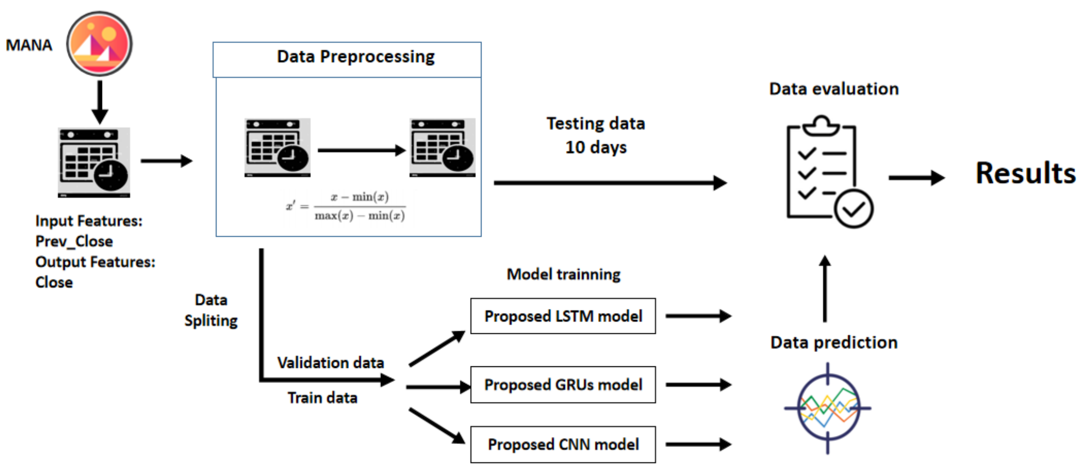


Figure 28: Experiments 1 Setup for MANA

4.5.2 Experiments 2 Setup

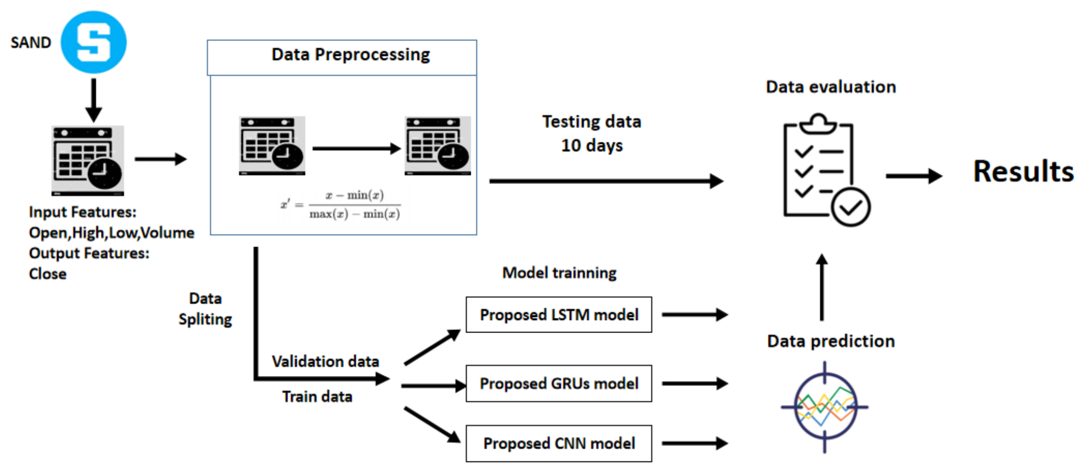


Figure 29: Experiments 2 Setup for SAND

Experiment 2 is set up like Figure 29 above for the SAND metaverse cryptocurrency. The dataset used is classical OHLCV datasets which are always used for the cryptocurrency trade. In this case, the input features are Open, High, Low and Volume, which means opening price, highest price, lowest price and total quantity of contracts traded of the time interval. The output feature and variable is Close, which means the closing price. Experiment 2 is to check the performance and efficiency of using Open, High, Low and Volume as an input variable to predict the next 10 days closing price of the metaverse cryptocurrencies. After processing the data preprocessing and feature selection, the data is split as 80-20 ratio, 80 % was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrency price and compare with the actual price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales. The same procedure will be repeated using SLP and MANA datasets shown in Figure 30 and Figure 31 below.

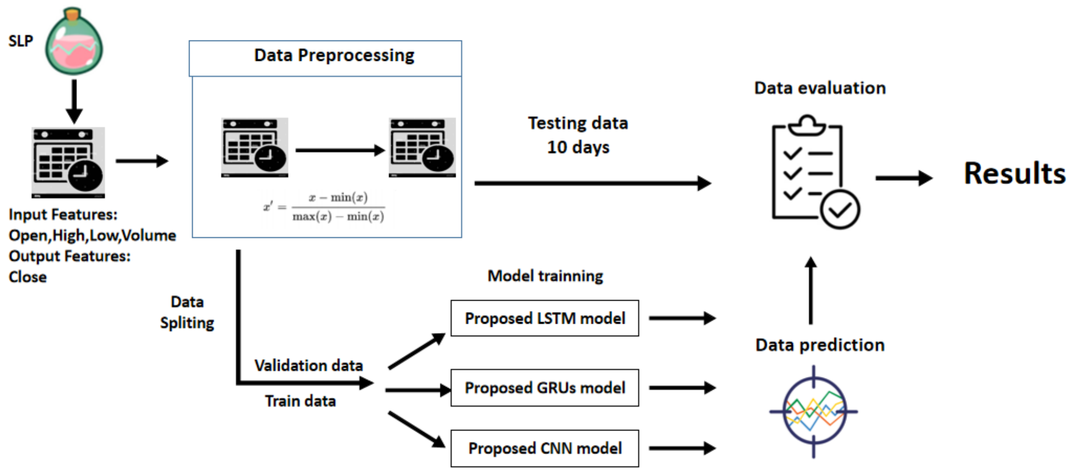


Figure 30: Experiments 2 Setup for SLP

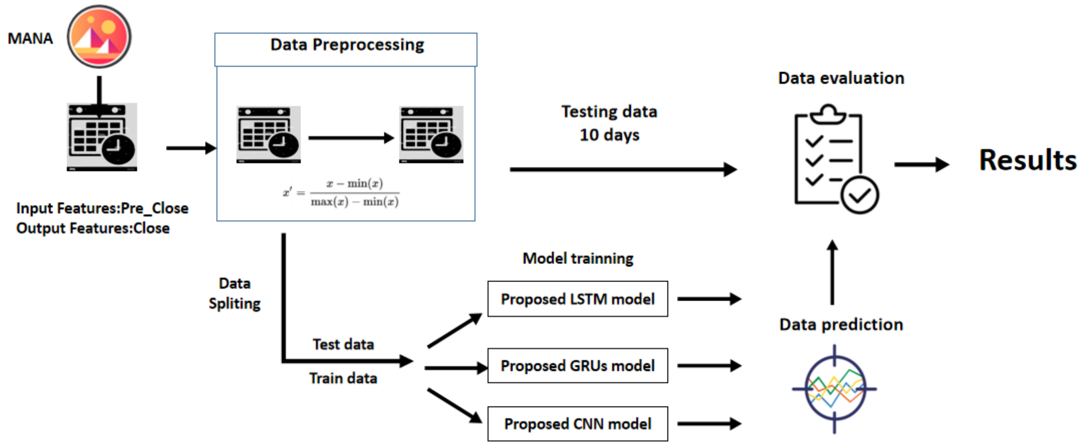


Figure 31: Experiments 2 Setup for MANA

4.5.3 Experiments 3 Setup

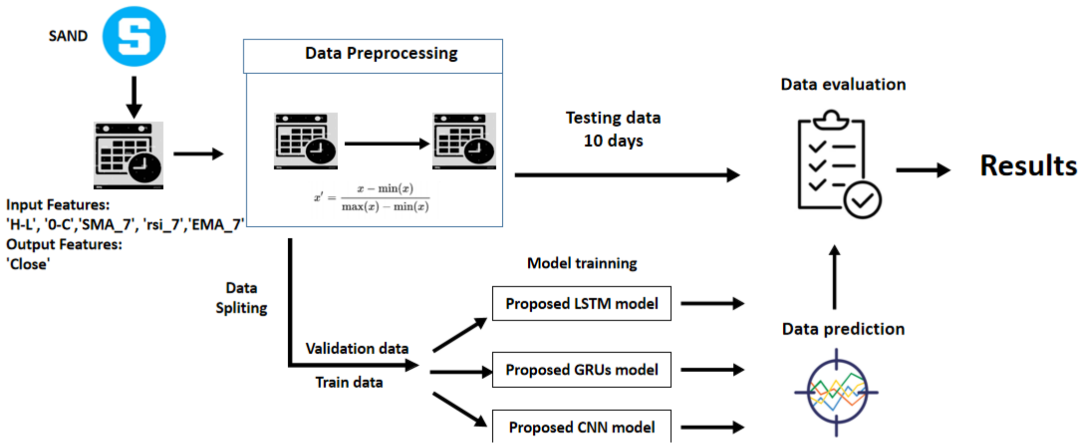


Figure 32: Experiments 3 Setup for SAND

Experiment 3 is set up like the Figure 32 above for the SAND metaverse cryptocurrency. The input features are differences between the highest and the lowest price at which a currency is traded on a particular trading day (H-L), differences between the first price and the last price at which a currency is traded on a particular trading day (O-C), the Simple Moving Average of past 7 days (SMA_7), the Exponential Moving Average of past 7 days (EMA_7) and the Relative Strength Index of past 7days (rsi_7). The output feature and variable is Close, which means the closing price. The importance of experiment 3 is to check the performance and efficiency of using these new generated technical indicators which are 'H-L', 'O-C', 'SMA_7', 'EMA_7' and 'rsi_7' as input variables to predict the next 10 days closing price of the metaverse cryptocurrencies. After processing the data preprocessing and feature selection, the data is split as 80-20 ratio, 80 % was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrencies price and compare with the actual cryptocurrencies price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales. The same procedure will be repeated using SLP and MANA datasets shown in Figure 33 and 34 below.

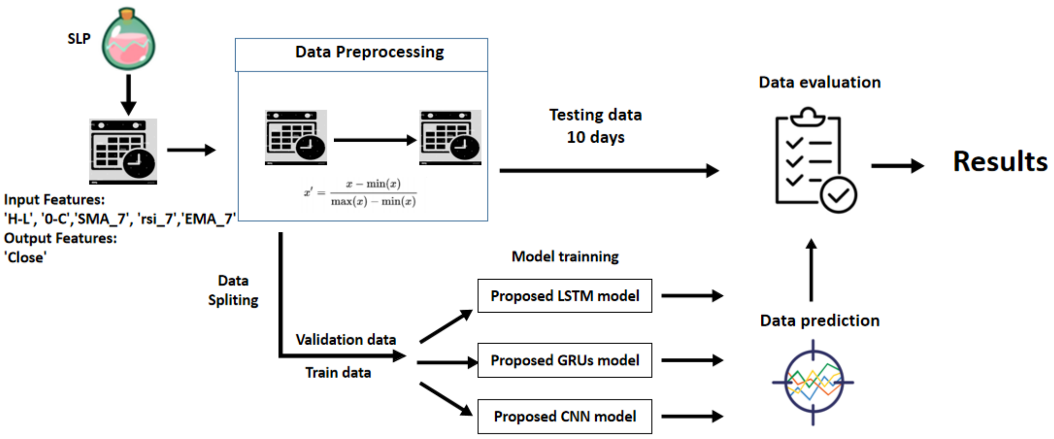


Figure 33: Experiments 3 Setup for SLP

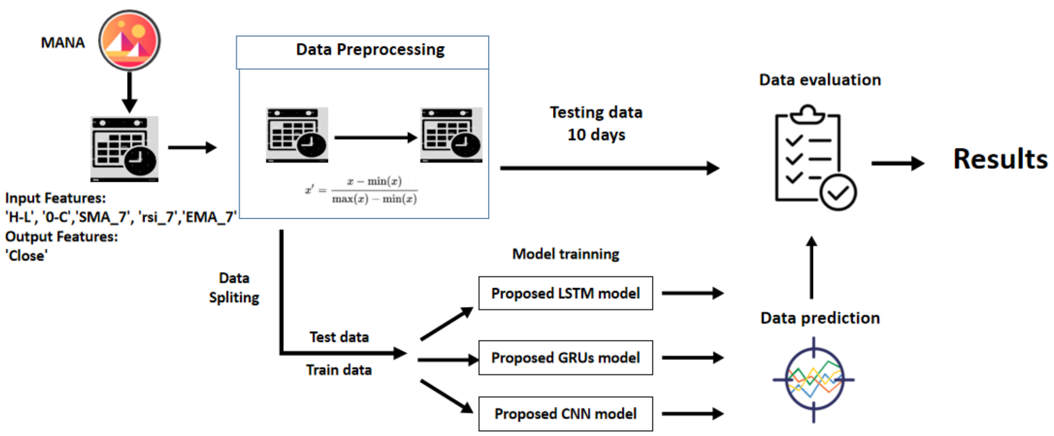


Figure 34: Experiments 3 Setup for MANA

4.5.4 Experiments 4 Setup

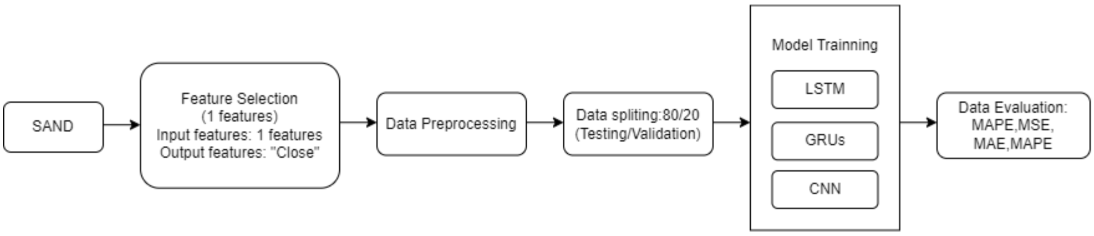


Figure 35:Experiments 4 Setup

Experiment 4 is set up like Figure 35 above for the SAND metaverse cryptocurrency. The input features using one feature from "Open", "High", "Low", "Volume", "Pre_Close", "H-L", "0-C", "SMA_7", "rsi_7", and "EMA 7". There are 10 test cases for experiment 4. The output feature "Close", which means the closing price. The importance of experiment 4 is to check the performance and efficiency of these ten dependent variables when feeding them individually without combination to predict the next 10 days' closing price of the metaverse cryptocurrencies by using LSTM, GRUs and CNN. After processing the data preprocessing and feature selection, the data is split an 80-20 ratio, 80 % was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrency price and compare it with the actual price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales.

4.5.5 Experiments 5 Setup

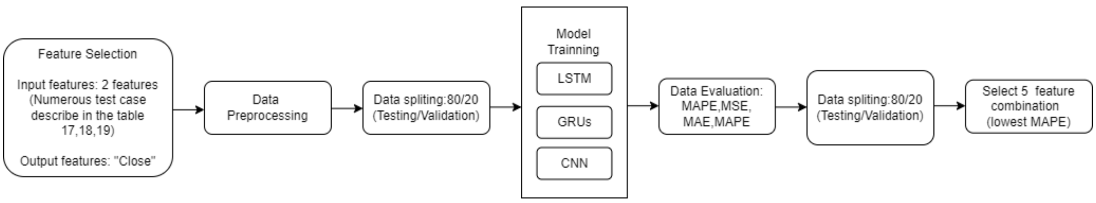


Figure 36: Experiments 5 Setup

Experiment 5 is set up like Figure 36 above for the SAND metaverse cryptocurrency. The input features using two dependent features from "Open", "High", "Low", "Volume", "Pre_Close", "H-L", "0-C", "SMA_7", "rsi_7", and "EMA 7". There are 45 test cases for experiment 5 that are tabulated in tables 17, 18 and 19 for LSTM, GRUs and CNN. The output feature "Close", which means the closing price. The importance of experiment 5 is to check the performance and efficiency of two features by testing every possible combination of the two features using the grid search method to predict the next ten days' closing price of the metaverse cryptocurrencies by using LSTM, GRUs and CNN. After processing the data preprocessing and feature selection, the data is split into an 80-20 ratio; 80% was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrency price and compare it with the actual price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales. The best top 5 two-feature combinations with the lowest MAPE values will be used for the following experiment: experiment 6.

4.5.6 Experiments 6 Setup

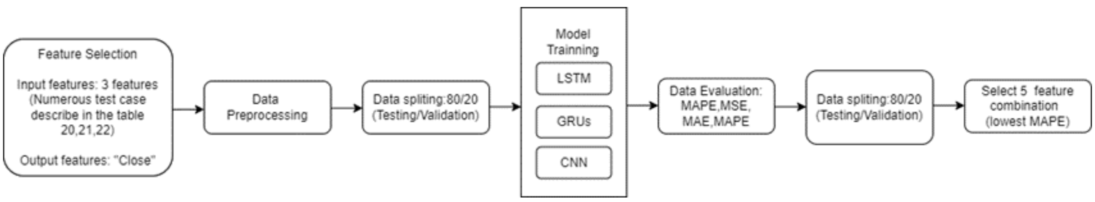


Figure 37: Experiments 6 Setup

Experiment 6 is set up like Figure 37 above for the SAND metaverse cryptocurrency. The input features are three feature-combinations using adding one more feature individually to the five selective two-feature combinations with the lowest MAPE value in experiment 5. There are 34, 36, and 40 test cases for experiment 5 that are tabulated in tables 20, 21 and 22 for LSTM, GRUs and CNN. The output feature

"Close", which means the closing price. The importance of experiment 6 is to check the performance and efficiency of three feature-combinations predict the next ten days' closing price of the metaverse cryptocurrencies by using LSTM, GRUs and CNN. After processing the data preprocessing and feature selection, the data is split into an 80-20 ratio; 80% was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrency price and compare it with the actual price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales. The best top 5 three-feature combinations with the lowest MAPE values will be used for the following experiment: experiment 7.

4.5.7 Experiments 7 Setup

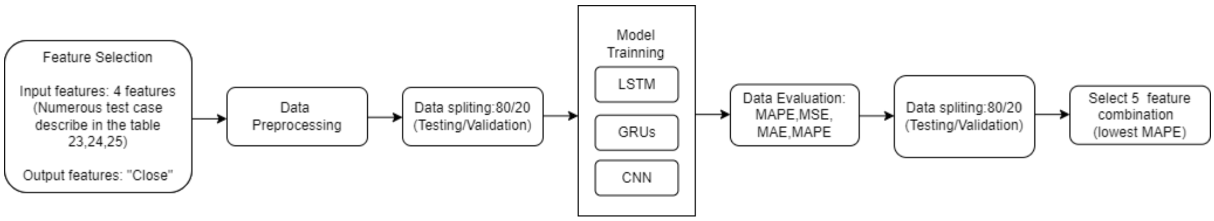


Figure 38: Experiments 7 Setup

Experiment 7 is set up like Figure 38 above for the SAND metaverse cryptocurrency. The input features are four feature-combinations using adding one more feature individually to the five selective three-feature combinations with the lowest MAPE value in experiment 6. In experiment 7, there are 32 test cases for LSTM, GRUs models and 33 test cases for CNN that are tabulated in tables 23, 24 and 25. The output feature "Close", which means the closing price. The importance of experiment 7 is to check the performance and efficiency of four feature-combinations to predict the next ten days' closing price of the metaverse cryptocurrencies by using LSTM, GRUs and CNN. After processing the data preprocessing and feature selection, the data is split into an 80-20 ratio; 80% was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrency price and compare it with the actual price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales. The best top 5 four-feature combinations with the lowest MAPE values will be used for the following experiment: experiment 8.

4.5.8 Experiments 8 Setup

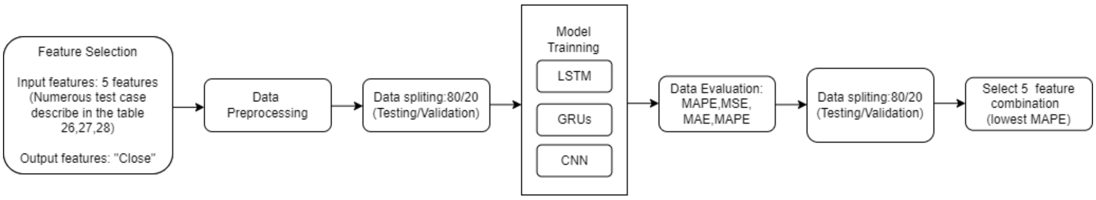


Figure 39: Experiments 8 Setup

Experiment 8 is set up like Figure 39 above for the SAND metaverse cryptocurrency. The input features are five feature-combinations using adding one more feature individually to the five selective four-feature combinations with the lowest MAPE value in experiment 7. In experiment 6, there are 26,29 and 25 test cases for LSTM, GRUs and CNN that are tabulated in tables 26, 27 and 28. The output feature "Close", which means the closing price. The importance of experiment 8 is to check the performance and efficiency of five feature-combinations to predict the next ten days' closing price of the metaverse cryptocurrencies by using LSTM, GRUs and CNN. After processing the data preprocessing and feature selection, the data is split into an 80-20 ratio; 80% was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrency price and compare it with the actual price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales. The best top 5 four-feature combinations with the lowest MAPE values will be used for the following experiment: experiment 9.

4.5.9 Experiments 9 Setup

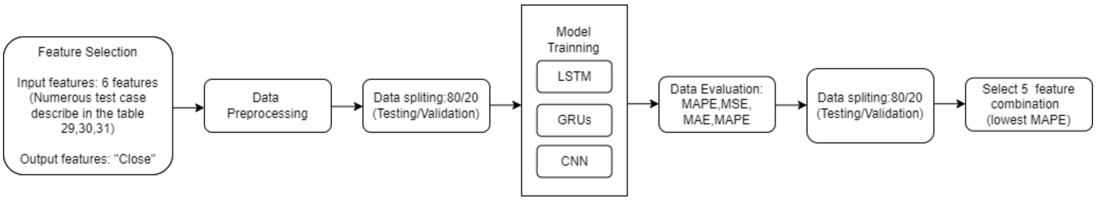


Figure 40: Experiment 9 Setup

Experiment 9 is set up like Figure 40 above for the SAND metaverse cryptocurrency. The input features are 6 feature-combinations using adding one more feature individually to the five selective five-feature combinations with the lowest MAPE value in experiment 8. There are 22,22, and 23 test cases for experiment 5 that are tabulated in tables 29, 30 and 31 for LSTM, GRUs and CNN. The output feature "Close", which means the closing price. The importance of experiment 9 is to check the performance and efficiency of 6 feature-combinations to predict the next ten days' closing price of the metaverse cryptocurrencies by using LSTM, GRUs and CNN. After processing the data preprocessing and feature selection, the data is split into an 80-20 ratio; 80% was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrency price and compare it with the actual price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales. The best top 5 six-feature combinations with the lowest MAPE values will be used for the following experiment: experiment 10.

4.5.10 Experiments 10 Setup

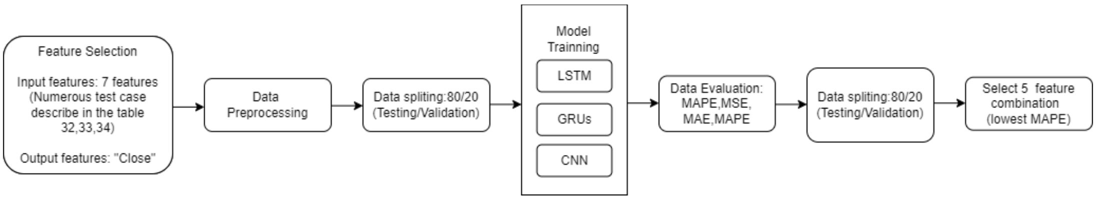


Figure 41: Experiments 10 Setup

Experiment 10 is set up like Figure 41 above for the SAND metaverse cryptocurrency. The input features are seven feature-combinations by adding one more feature individually to the five selective six-feature combinations with the lowest MAPE value in experiment 9. There are 13, 16 and 17 test cases for experiment 10 that are tabulated in tables 32, 33 and 34 for LSTM, GRUs and CNN. The output feature "Close", which means the closing price. The importance of experiment 10 is to check the performance and efficiency of seven feature-combinations to predict the next ten days' closing price of the metaverse cryptocurrencies by using LSTM, GRUs and CNN. After processing the data preprocessing and feature selection, the data is split into an 80-20 ratio; 80% was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrency price and compare it with the actual price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales. The best top 5 seven-feature combinations with the lowest MAPE values will be used for the following experiment: experiment 11.

4.5.11 Experiments 11 Setup

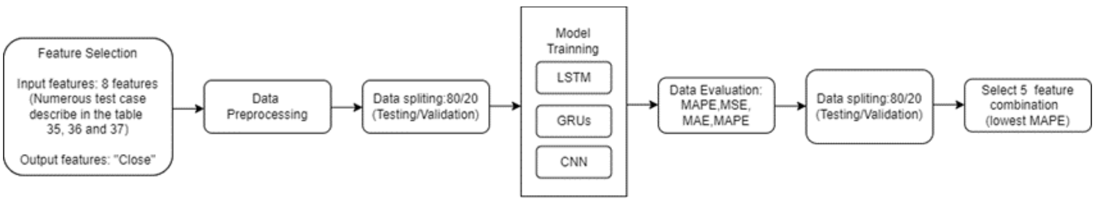


Figure 42: Experiments 11 Setup

Experiment 11 is set up like Figure 42 above for the SAND metaverse cryptocurrency. The input features are eight feature-combinations by adding one more feature individually to the five selective seven-feature combinations with the lowest MAPE value in experiment 10. There are 8 test cases for LSTM and GRUs and 12 test cases for CNN in experiment 10 that are tabulated in tables 35, 36 and 37. The output feature "Close", which means the closing price. The importance of experiment 10 is to check the performance and efficiency of eight feature-combinations to predict the next ten days' closing price of the metaverse cryptocurrencies by using LSTM, GRUs and CNN. After processing the data preprocessing and feature selection, the data is split into an 80-20 ratio; 80% was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrency price and compare it with the actual price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales. The best top 5 eight-feature combinations with the lowest MAPE values will be used for the following experiment: experiment 12.

4.5.12 Experiments 12 Setup

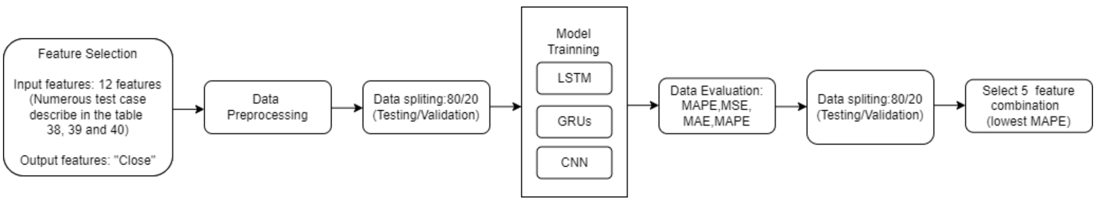


Figure 43: Experiments 12 Setup

Experiment 12 is set up like Figure 43 above for the SAND metaverse cryptocurrency. The input features are nine feature-combinations by adding one more feature individually to the five selective eight-feature combinations with the lowest MAPE value in experiment 11. There are 6 test cases for experiment 12 that are tabulated in tables 38,39 and 40 for LSTM, GRUs and CNN. The output feature "Close", which means the closing price. The importance of experiment 12 is to check the performance and efficiency of nine feature-combinations to predict the next ten days' closing price of the metaverse cryptocurrencies by using LSTM, GRUs and CNN. After processing the data preprocessing and feature selection, the data is split into an 80-20 ratio; 80% was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrency price and compare it with the actual price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales.

4.5.13 Experiments 13 Setup

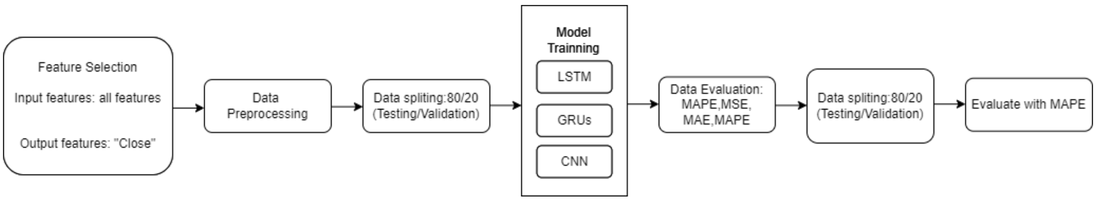


Figure 44: Experiments 13 Setup

Experiment 13 is set up like Figure 44 above for the SAND metaverse cryptocurrency. The input features use all variables available in the dataset. There are only 1 test cases for experiment 13 that are tabulated in tables 41 for LSTM, GRUs and CNN. The output feature "Close", which means the closing price. The importance of experiment 13 is to check the performance and efficiency of all feature-combinations to predict the next ten days' closing price of the metaverse cryptocurrencies by using LSTM, GRUs and CNN. After processing the data preprocessing and feature selection, the data is split into an 80-20 ratio; 80% was for the training purpose, and the remaining 20 % was for validation purposes and fed into three different models, which are proposed LSTM, GRUs and CNN models. The last ten days for the input datasets will be input to the models to predict the next ten days of the metaverse cryptocurrency price and compare it with the actual price, which is labelled as testing data. Finally, the result is evaluated with standard performance metrics, which are

MSE, MAE, MAPE and RMSE and tabulated into tables. MAPE will be the primary metric as MAPE is the absolute error normalized over the data, which allows the error to be compared across data with different scales.

4.6 Hyperparameter Experiments Setup

In experiments 1 to 3, we compared the performance of three different machine learning models (LSTM, GRUs, and CNN) on a dataset that included various features such as "Pre_close", "Primary-technical indicator", and "Advance technical indicator". To create the most accurate predictive model, we also added additional test cases based on features that were identified as having the lowest mean absolute percentage error (MAPE) in a previous experiment involving a range of features (experiments 4 to 13). For the LSTM model, the additional test case consists of the feature "Volume" for the GRUs model the additional test case includes the features "Open, Low, Volume, H-L, 0-C, rsi_7" and for the CNN model the additional test case includes the features "rsi_7, EMA_7". As a result, there will be a total of 12 test cases, with four test cases for each of the three models (LSTM, GRUs, and CNN) that will be used as the input features for the experiments 14 to 20.

4.6.1 Experiments 14 Setup

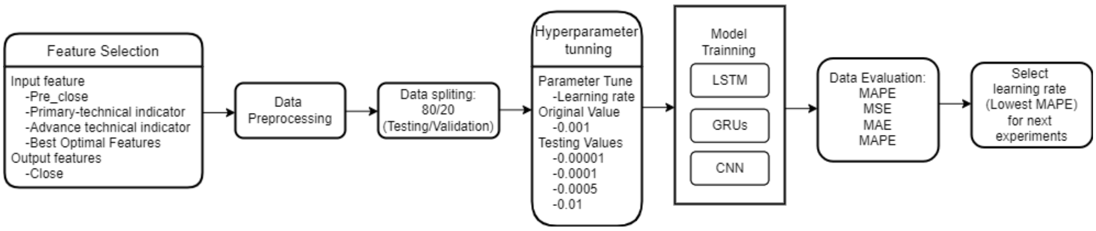


Figure 45: Experiments 14 Setup

The experiment 14 uses the test case above which are "Pre_close", "Primary-technical indicator", "Advance technical indicator", " Best Optimal Features " as the input features to predict the next ten days' closing price of a metaverse cryptocurrency using LSTM, GRUs, and CNN models. The output feature also remains the same which is "Close", the closing price of the SAND cryptocurrencies. The data is split 80/20 for training and validation, and the models are tested on the last ten days of the input data to predict the next ten days of cryptocurrency prices. To further optimize the models, we will also perform hyperparameter tuning by testing different learning rate values (0.00001, 0.0001, 0.0005, 0.01) in place of the default value of 0.001. The results are evaluated using MSE, MAE, MAPE, and RMSE and are tabulated. MAPE is the primary metric as it allows for comparison of error across data with different scales.

4.6.2 Experiments 15 Setup

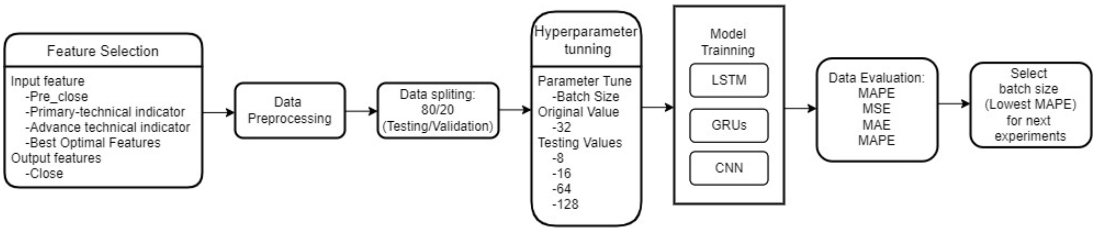


Figure 46: Experiments 15 Setup

The experiment 15 uses the test case above which are "Pre_close", "Primary-technical indicator", "Advance technical indicator", " Best Optimal Features " as the input features to predict the next ten days' closing price of a metaverse cryptocurrency using LSTM, GRUs, and CNN models. The output feature also remains the same which is "Close", the closing price of the SAND cryptocurrencies. The data is split 80/20 for training and validation, and the models are tested on the last ten days of the input data to predict the next ten days of cryptocurrency prices. To further optimize the models, we will also perform hyperparameter tuning by testing different batch size values (8,16,64,128) in place of the default value of 32. The results are evaluated using MSE, MAE, MAPE, and RMSE and are tabulated. MAPE is the primary metric as it allows for comparison of error across data with different scales.

4.6.3 Experiments 16 Setup

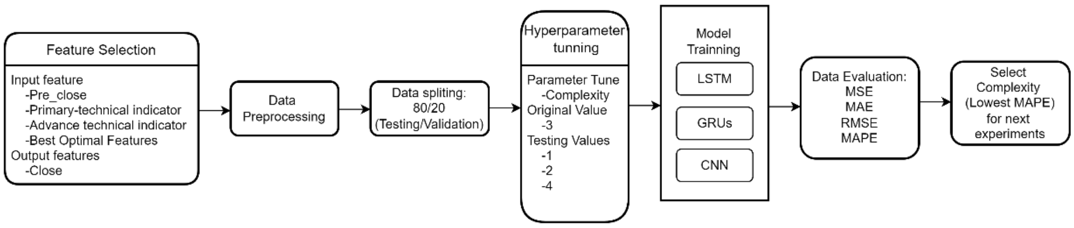


Figure 47: Experiments 16 Setup

The experiment 16 uses the test case above which are "Pre_close", "Primary-technical indicator", "Advance technical indicator", " Best Optimal Features " as the input features to predict the next ten days' closing price of a metaverse cryptocurrency using LSTM, GRUs, and CNN models. The output feature also remains the same which is "Close", the closing price of the SAND cryptocurrencies. The data is split 80/20 for training and validation, and the models are tested on the last ten days of the input data to predict the next ten days of cryptocurrency prices. To further optimize the models, we will also perform hyperparameter tuning by testing different complexity (1,2,4) that will be fully described in Table 53,in place of the default value of 4. The results are evaluated using MSE, MAE, MAPE, and RMSE and are tabulated. MAPE is the primary metric as it allows for comparison of error across data with different scales.

4.6.4 Experiments 17 Setup

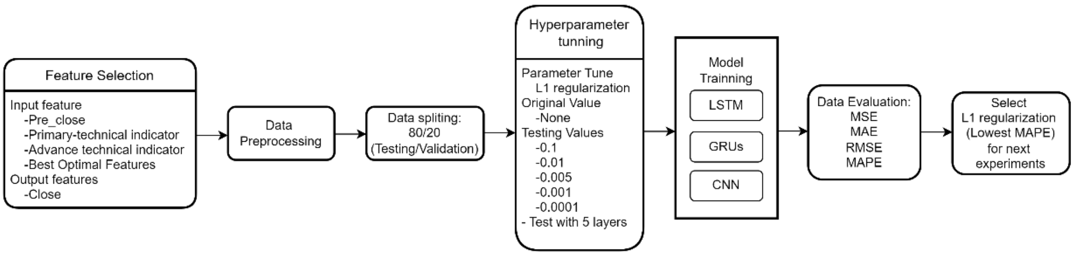


Figure 48: Experiments 17 Setup

The experiment 17 uses the test case above which are "Pre_close", "Primary-technical indicator", "Advance technical indicator", " Best Optimal Features " as the input features to predict the next ten days' closing price of a metaverse cryptocurrency using LSTM, GRUs, and CNN models. The output feature also remains the same which is "Close", the closing price of the SAND cryptocurrencies. The data is split 80/20 for training and validation, and the models are tested on the last ten days of the input data to predict the next ten days of cryptocurrency prices. To further optimize the models, we will also perform hyperparameter tuning by testing different L1 Regularization (0.1, 0.01, 0.005, 0.001,0.0001) with different multilayer of the architecture of LSTM, GRUs and CNN model)that will be fully described in Table 53, in place of the default value of 4. The results are evaluated using MSE, MAE, MAPE, and RMSE and are tabulated. MAPE is the primary metric as it allows for comparison of error across data with different scales.

4.6.5 Experiments 18 Setup

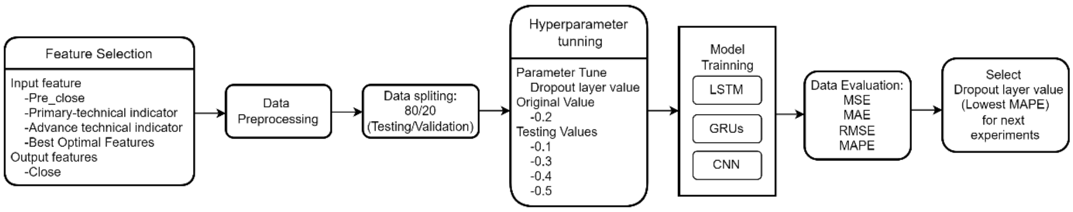


Figure 49: Experiments 18 Setup

The experiment 18 uses the test case above which are "Pre_close", "Primary-technical indicator", "Advance technical indicator", " Best Optimal Features " as the input features to predict the next ten days' closing price of a metaverse cryptocurrency using LSTM, GRUs, and CNN models. The output feature also remains the same which is "Close", the closing price of the SAND cryptocurrencies. The data is split 80/20 for training and validation, and the models are tested on the last ten days of the input data to predict the next ten days of cryptocurrency prices. To further optimize the models, we will also perform hyperparameter tuning by testing different dropout layer value (0.1, 0.3, 0.4, 0.5) with different multilayer of the architecture of LSTM, GRUs and CNN model) in place of the default value of 0.2. The results are evaluated using MSE, MAE, MAPE, and RMSE and are tabulated. MAPE is the primary metric as it allows for comparison of error across data with different scales.

4.6.6 Experiments 19 Setup

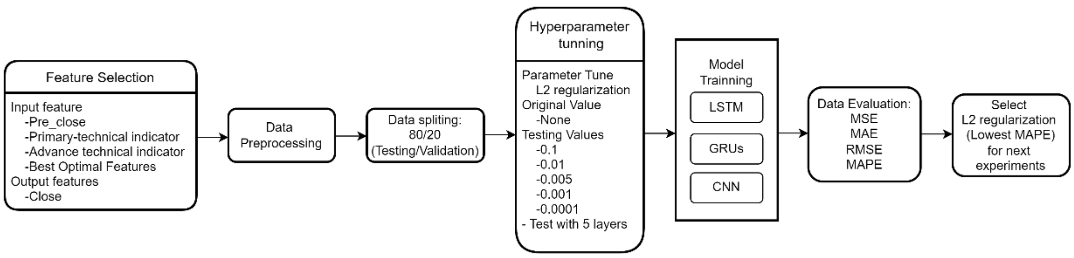


Figure 50: Experiments 19 Setup

The experiment 19 uses the test case above which are "Pre_close", "Primary-technical indicator", "Advance technical indicator", " Best Optimal Features " as the input features to predict the next ten days' closing price of a metaverse cryptocurrency using LSTM, GRUs, and CNN models. The output feature also remains the same which is "Close", the closing price of the SAND cryptocurrencies. The data is split 80/20 for training and validation, and the models are tested on the last ten days of the input data to predict the next ten days of cryptocurrency prices. To further optimize the models, we will also perform hyperparameter tuning by testing different L2 Regularization (0.1, 0.01, 0.005, 0.001,0.0001) with different multilayer of the architecture of LSTM, GRUs and CNN model)that will be fully described in Table 53, in place of the default value of 4. The results are evaluated using MSE, MAE, MAPE, and RMSE and are tabulated. MAPE is the primary metric as it allows for comparison of error across data with different scales.

4.6.7 Experiments 20 Setup

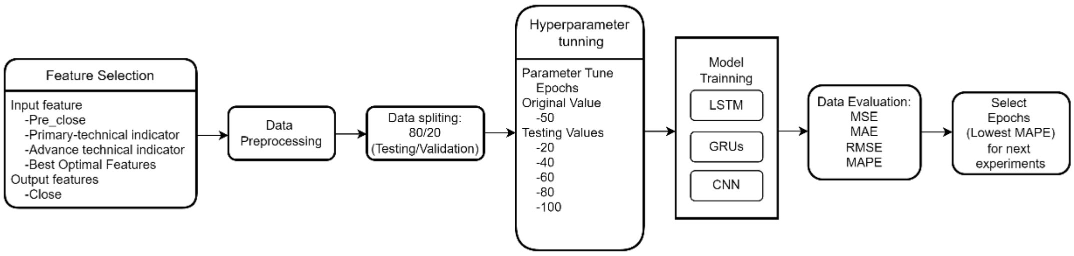


Figure 51: Experiments 20 Setup

The experiment 20 uses the test case above which are "Pre_close", "Primary-technical indicator", "Advance technical indicator", " Best Optimal Features " as the input features to predict the next ten days' closing price of a metaverse cryptocurrency using LSTM, GRUs, and CNN models. The output feature also remains the same which is "Close", the closing price of the SAND cryptocurrencies. The data is split 80/20 for training and validation, and the models are tested on the last ten days of the input data to predict the next ten days of cryptocurrency prices. To further optimize the models, we will also perform hyperparameter tuning by testing different L2 Regularization (20, 40, 60, 80, 100) with in place of the default value of 50. The results are evaluated using MSE, MAE, MAPE, and RMSE and are tabulated. MAPE is the primary metric as it allows for comparison of error across data with different scales.

4.7 Conclusion

In conclusion, chapter 4, experimental design, describes and shows the flow of experiments 1 to 20 in diagram form. The experimental design is explained explicitly throughout the chapter, including data and feature engineering used in experiments 1 to 20, the model designs and architecture of the proposed LSTM, GRUs and CNN model, the model evaluation metrics, and the setup of the experiment, including the parameter setup, dependent and targets variables of both three experiments.

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

This chapter discusses the implementation of the Python code by using the Google Colab as the implementation environment and Python 3 as the implementation language. This section discusses the implementation of feature selection experiments(1-13) and hyperparameter tuning experiments(14-20).In feature selection experiments, experiment 1 investigates the performance of using the "Previous Close" feature to predict the next ten days of SAND, SLP and MANA, metaverse cryptocurrency closing price feature in terms of MAPE, MSE, MAPE and RMSE by using three models, which are LSTM, GRUs and CNN. Experiment 2 is to investigate the performance of using "Open", "High", "Low" and "Volume" features to predict the next ten days of "Close" of SAND, SLP and MANA, metaverse cryptocurrencies closing price feature in metrics of MAPE, MSE, MAE and RMSE. Experiment 3 is to investigate the performance of using "Pre_Close", "H-L," "0-C", "SMA_7", "rsi_7", "EMA_7" features to predict the next 10 days of "Close" of SAND, SLP and MANA, metaverse cryptocurrencies closing price feature in metrics of MAPE, MSE, MAPE and RMSE. Experiments 4 to 13 is to investigate the performance of using different feature combinations (1 feature to 10 features) to predict the next 10 days of "Close" of SAND, SLP and MANA, metaverse cryptocurrencies closing price feature in metrics of MAPE, MSE, MAPE and RMSE. Experiments 4 to 13 is to investigate the performance of using different feature combinations (1 feature to 10 features) to predict the next 10 days of "Close" of SAND, SLP and MANA, metaverse cryptocurrencies closing price feature in metrics of MAPE, MSE, MAPE and RMSE.

5.2 Experiment 1 Implementation

Experiment 1 investigates the performance of using the 'Previous Close' feature to predict the next ten days of SAND, SLP, and MANA, metaverse cryptocurrency closing price feature in terms of MAPE, MSE, MAPE and RMSE by using three models, which are LSTM, GRUs, and CNN. In 5.1 sections show the implementations of the proposed LSTM, GRUs and CNN by using SAND data sets. Similarly, the whole steps in the 5.1 section need to repeat using SLP and MANA datasets by altering the pd.read_csv(sand) function to pd.read_csv(slp) and pd.read_csv(mana) separately to predict both closing prices.

5.2.1 The Proposed Lstm Model

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
from sklearn.preprocessing import MinMaxScaler
import sklearn.metrics as metrics
import ta
```

Figure 52: Code Snippet for Importing Library

At first, we need to import the library that is useful for data pre-processing showed in Figure 52, such as NumPy, Pandas, Matplotlib, Math, Ta and Sklearn. Sklearn library is used for the data normalization and evaluation metrics for performances. Ta is a technical analysis library useful for feature engineering from financial time series datasets (Open, Close, High, Low, and Volume).

```
sand = 'https://raw.githubusercontent.com/loguansiang/fyp/main/SAND%20fyp.csv'
slp= 'https://raw.githubusercontent.com/loguansiang/fyp/main/SLP%20fyp.csv'
mana = 'https://raw.githubusercontent.com/loguansiang/fyp/main/MANA%20fyp.csv'


df = pd.read_csv(sand)
df['Date'] = pd.to_datetime(df['Date'])
df.set_index('Date', inplace = True)
```

Figure 53: Code Snippet for Importing Dataset

The Sand, SLP and MANA datasets downloaded from Yahoo Finance in CSV format have been uploaded to GitHub. We import the sand, SLP and mana datasets through the GitHub link shown in Figure 53. The column 'Date' is converted string Date time into Python Date time object and set as the index of a Data Frame.

```
df['Pre_Close'] = df['Close'].shift(+1)
df[('H-L')] = df['High'] - df['Low']
df[('0-C')]= df['Open'] - df['Close']
df['SMA_7'] = df.Close.rolling(7, min_periods=1).mean()
df['rsi_7']=ta.momentum.rsi(df.Close,window=7)
df['EMA_7']= df['Close'].ewm(span=7).mean()
df = df.drop(columns=['Adj Close'])
df.dropna(inplace= True)
```

Figure 54: Code Snippet for Feature Engineering

The new data frame is generated by adding the 6 new features shown in Figure 54: "Pre_Close", "H-L", "0-C", "SMA_7", "rsi_7" and "EMA_7". The "Adj Close" is eliminated from the original data frame.

```
def data_prep(dataset, target, data_start, data_end, look_back, horizon):
    dataX,dataY = [],[]
    data_start = data_start + look_back
    if data_end is None:
        data_end = len(dataset) - horizon

    for i in range(data_start, data_end):
        a = range(i-look_back, i)
        dataX.append(dataset[a])
        b = range(i+1, i+1+horizon)
        dataY.append(target[b])
```

```
    return np.array(dataX), np.array(dataY)
```

Figure 55: Code Snippet for Data Preprocessing

This data_preprocessing function shown in Figure 55 takes five arguments: dataset, which is a NumPy array that we named as the input feature, the target, which is a NumPy array that we named as the target feature, data_start, which is the start index where the dataset is input, data_end which the end index where the dataset is input, look_back which is the number of previous time steps to use as input variables to predict the following timeframes and the last horizon is the prediction time stamp. This default will create a dataset where dataX is the selected feature value at the given time and dataY is the closing price of the cryptocurrency at the next horizon.

```
X_scaler = MinMaxScaler()
Y_scaler = MinMaxScaler()
X_data = X_scaler.fit_transform(df[['Pre_Close']])
Y_data = Y_scaler.fit_transform(df[['Close']])
```

Figure 56: Code Snippet for Feature Selection

X_scaler and Y_scaler are responsible for doing the min-max normalization based on the feature selected for X_data and Y_data shown in Figure 56. In experiment 1, for the SAND, "Pre_Close" is used as the input feature and variables, and the "Close" is set as the target feature and variable.

```
look_back = 10
horizon = 10
train_split = int(len(df) * 0.8)
x_train, y_train = data_prep(X_data, Y_data, 0, train_split, look_back, horizon)
x_vali, y_vali = data_prep(X_data, Y_data, train_split, None, look_back, horizon)
```

Figure 57: Code Snippet for Train-validation Split

The look_back and horizon are set as ten days in Figure 57, and the train_split is set as 80 percent as train-set and the remaining 20 percent as the validation set. X_train represents the train set for the "Pre_Close", y_train represents the train set for the"Close", x_valid represents the train set for the "Pre_Close", and y_valid represents the train set for the "Close".

```
train_data = tf.data.Dataset.from_tensor_slices((x_train, y_train))
train_data = train_data.batch(batch_size).repeat()
val_data = tf.data.Dataset.from_tensor_slices((x_vali, y_vali))
val_data = val_data.batch(batch_size).repeat()
```

Figure 58: Code Snippet for Prepare Train and Validation Data

In Figure 58, the TensorFlow data function is used to prepare the training and validation data, which is a faster and more efficient way to feed data for training the model.

```
import tensorflow as tf
from tensorflow.keras import layers
# Set random seed for as reproducible results as possible
tf.random.set_seed(7)
lstm_model = Sequential()
lstm_model.add(LSTM(100,return_sequences=True,input_shape=(x_train.shape[-2:])))
lstm_model.add(Dense(20,activation='linear'))
lstm_model.add(LSTM(50))
lstm_model.add(Dropout(0.2))
lstm_model.add(Dense(units=horizon,activation='linear'))
lstm_model.add(Activation('linear'))
lstm_model.compile(loss='mean_squared_error', optimizer='adam')
lstm_model.summary()
history = lstm_model.fit(train_data,epochs=50,steps_per_epoch=100,validation_data=val_
data,validation_steps=50,verbose=1)
```

Figure 59: Build and Compile the LSTM Model

In Figure 59, the proposed LSTM model is built and compiled by using the parameter of epoch 50, linear activation layers, mean square error as loss function and adam optimizer. For the proposed LSTM model's architecture, we follow the architecture already stated in Figure 22. The proposed LSTM models consist of 100 LSTM cells input layers of 10 observations, followed by a dense layer of 20 output shapes, 50 LSTM cells layers, 0.2 dropout layers, 10 output shape dense layers and a linear activation layer. After the model compiles, we fed the train and validation data into the model we built.

```
data_val = X_scaler.fit_transform(df[['Pre_Close']].tail(10))
val_rescaled = data_val.reshape(1, data_val.shape[0], data_val.shape[1])
pred =lstm_model.predict(val_rescaled)
pred_Inverse = Y_scaler.inverse_transform(pred)
pred_Inverse
```

Figure 60: Code Snippet for Using LSTM Model to Predict Closing Price

We prepare the testing data for the last ten days of Pre_Close values into the data_val variable shown in Figure 60. We reshape the size of the data_val variable and fit it into the proposed LSTM model to predict the next 10 days of the "Close" values. The pred values need to inverse to the feature range of "Close" by using Y_Scaler inverse_transform function and stored inside the pred_inverse variable.

```
def timeseries_evaluation_metrics_func(y_true, y_pred):
    def mape(y_true, predictions):
        y_true, predictions = np.array(y_true), np.array(predictions)
        return np.sum(np.abs(y_true - predictions)/np.sum(y_true)*100)
    print('Evaluation metric results:-')
    print(f'MSE is : {metrics.mean_squared_error(y_true, y_pred)}')
    print(f'MAPE% is : {mape(y_true, y_pred)}')
    print(f'RMSE is : {np.sqrt(metrics.mean_squared_error(y_true, y_pred))}')
   print(f'MAPE is : {mean_absolute_percentage_error(y_true, y_pred)}')
validate = (df[['Close']].tail(10))
```

```
timeseries_evaluation_metrics_func(validate['Close'],pred_Inverse)
```

Figure 61: Code Snippet for Model Evaluation

Finally, evaluate the result with standard performance metrics using code in Figure 61. The evaluated results and analysis will be tabulated in table forms to compare with another two models, GRUs and CNN, and discussed in chapter 6 later. The actual value labelled as y_true is the last ten rows of the "Close" data that have been represented by validate["Close"], which does not fit into the proposed model. The predicted value labeled as y_pred is the predicted value labeled as the pred_inverse variable.

5.2.2 The Proposed GRU Model

For the GRUs model, we first need to repeat the steps stated in the code snippet from Figure 52 to Figure 58. The steps include importing the library, dataset, feature engineering, pre-processing, feature selection, train-validation split, and using the TensorFlow data function to prepare train and validation data.

```
import tensorflow as tf
from tensorflow.keras import layers
# Set random seed for as reproducible results as possible
tf.random.set_seed(7)
gru_model = Sequential()
gru_model.add(GRU(100,return_sequences=True,input_shape=(x_train.shape[-2:])))
gru_model.add(Dense(20,activation='linear'))
gru_model.add(GRU(50))
gru_model.add(Dropout(0.2))
gru_model.add(Dense(units=horizon,activation='linear'))
gru_model.add(Activation('linear'))
gru_model.compile(loss='mean_squared_error', optimizer='adam')
gru_model.summary()
history = gru_model.fit(train_data,epochs=50,steps_per_epoch=100,validation_data=val_d
ata,validation_steps=50,verbose=1)
```

Figure 62: Build and Compile the GRUs Model

The use of the set.seed function is to ensure that we get the same results for randomization. We build and compile the proposed GRUs model mentioned in Figure 62.The proposed GRUs model is built and compiled by using the parameter of epoch 50, linear activation layers, mean square error as loss function and adam optimizer. For the proposed GRUs model's architecture, we follow the architecture already stated in Figure 23. The proposed GRUs models consist of 100 GRUs cells input layers of 10 observation, followed by a dense layer of 20 output shape, 50 GRUs cells layer, 0.2 dropout layer, 10 output shape dense layer and a linear activation layer. After that, we configured the model and started training the proposed model. We also plot out the loss curve by training and validation process to measure the prediction model's efficiency in predicting the expected outcome.

```
data_val = X_scaler.fit_transform(df[['Pre_Close']].tail(10))
val_rescaled = data_val.reshape(1, data_val.shape[0], data_val.shape[1])
pred =gru_model.predict(val_rescaled)
pred_Inverse = Y_scaler.inverse_transform(pred)
validate = (df[['Close']].tail(10))
timeseries_evaluation_metrics_func(validate['Close'],pred_Inverse[0])
```

Figure 63: Code Snippet for Using GRU Model to Predict Closing Price

We prepare the testing data for the last ten days of Pre_Close values into the data_val variable. We reshape the size of the data_val variable and fit it into the proposed GRUs model to predict the next 10 days of the "Close" values. The pred values need to inverse to the feature range of "Close" by using Y_Scaler inverse_transform function and stored inside the pred_inverse variable. Finally, evaluate the result with standard performance metrics by fitting the "timeseries_evaluation_metrics_func" functions in Figure 63 for model evaluation. The evaluated results and analysis will be tabulated in table forms to compare with another two models, LSTMs and CNN, and discussed in chapter 6 later. The actual value labelled as y_true is the last ten rows of the "Close" data that have been represented by validate["Close"], which does not fit into the proposed model. The predicted value labelled as y_pred is the predicted value labelled as the pred_inverse variable.

5.2.3 The Proposed CNN Model

For the CNN model, we first need to repeat the steps stated in the code snippet (Figure 52 to Figure 58). The steps include importing the library, dataset, feature engineering, data pre-processing, feature selection, train-validation split, and TensorFlow data function to prepare train and validation data.

```
# define model
import tensorflow as tf
from tensorflow.keras import layers
# Set random seed for as reproducible results as possible
tf.random.set_seed(7)
cnn_model = Sequential()
cnn_model.add(Conv1D(filters=32, kernel_size=1, activation='relu', input_shape=(x_train.s
hape[-2:])))
cnn_model.add(Conv1D(filters=64, kernel_size=1))
cnn_model.add(MaxPooling1D(pool_size=(1)))
cnn_model.add(Conv1D(filters=128, kernel_size=1))
cnn_model.add(Flatten())
cnn_model.add(Dense(256,))
cnn_model.add(Dropout(0.2))
cnn_model.add(Dense(10))
cnn_model.compile(optimizer='adam', loss='mse')
cnn_model.summary()
history = cnn_model.fit(train_data,epochs=50,steps_per_epoch=100,validation_data=val_d
ata,validation_steps=50,verbose=1)
```

Figure 64: Build and Compile the CNN Model

The use of the set.seed function is to ensure that we get the same results for randomization. We build and compile the proposed CNN model mentioned in Figure

64. The proposed CNN model consists of a Conv1D layer with a filter size of 32, kernel size of 1, relu activation layer and 10 observations as input shape, followed by a Conv1D layer with a filter size of 64, kernel size of 1, max pooling layer with pooling size of 1, Conv1D layer with a filter size of 128, kernel size of 1,1 flatten layer and a fully connected layer of 2 dense layers with output shape 256 and 10 and one 0.2 size dropout layer between them. After that, we configured the model and started training the proposed model. We also plot out the loss curve by training and validation process to measure the prediction model's efficiency in predicting the expected outcome.

```
data_val = X_scaler.fit_transform(df[['Pre_Close']].tail(10))
val_rescaled = data_val.reshape(1, data_val.shape[0], data_val.shape[1])
pred =cnn_model.predict(val_rescaled)
pred_Inverse = Y_scaler.inverse_transform(pred)
pred_Inverse
```

Figure 65: Code Snippet for Using GRU model to predict Closing Price

We prepare the testing data for the last 10 days of Pre_Close values into the data_val variable. Then, we reshape the size of data_val variable and fit it into the proposed CNNs model to predict the next 10 days of the 'Close' values. The pred values need to inverse to the feature range of 'Close' by using Y_Scaler inverse_transform function and stored inside the pred_inverse variable. Finally, evaluate the result with standard performance metrics by fitting the 'timeseries_evaluation_metrics_func' functions in Figure 65 for model evaluation. The evaluated results and analysis will be tabulated in table forms to compare with another two models, LSTMs and GRUs, discussed in chapter 6 later. The actual value labelled as y_true is the last ten rows of the 'Close' data that have been represented by validate["Close"], which does not fit into the proposed model. The predicted value labelled as y_pred is the predicted value labelled as the pred_inverse variable.

5.3 Experiment 2 Implementation

Experiment 2 is to investigate the performance of using the "Open', 'High', 'Low', 'Volume' feature to predict the next ten days of SAND, SLP and MANA, metaverse cryptocurrency closing price feature in terms of MAPE, MSE, MAPE, and RMSE by using three models which are LSTM, GRUs, and CNN. In 5.3 sections show the implementations of the proposed LSTM, GRUs, and CNN by using SAND data sets. Similarly, the whole steps in the 5.3 section need to repeat using SLP and mana datasets by altering the pd.read_csv(sand) function to pd.read_csv(slp) and pd.read_csv(mana) separately to predict both closing prices.

5.3.1 The Proposed LSTM Model

For the LSTM model, the experiment 2 flow is principally the same as experiment 1, the code snippet, which is importing the library, importing the dataset of SAND metaverse cryptocurrencies, feature engineering, and data preprocessing, is repeated the same as the experiment 1, as the library used, the dataset is the same. It can be observed in the code snippet Figure 66 below.

```
X_scaler = MinMaxScaler()
Y_scaler = MinMaxScaler()
X_data = X_scaler.fit_transform(df[['Open', 'High', 'Low','Volume']])
Y_data = Y_scaler.fit_transform(df[['Close']])
look_back = 10
horizon = 10
train_split = int(len(df) * 0.8)
batch_size = 32
x_train, y_train = data_prep(X_data, Y_data, 0, train_split, look_back, horizon)
x_vali, y_vali = data_prep(X_data, Y_data, train_split, None, look_back, horizon)
train_data = tf.data.Dataset.from_tensor_slices((x_train, y_train))
train_data = train_data.batch(batch_size).repeat()
val_data = tf.data.Dataset.from_tensor_slices((x_vali, y_vali))
val_data = val_data.batch(batch_size).repeat()
tf.random.set_seed(7)
lstm_model = Sequential()
```

```
lstm_model.add(LSTM(100,return_sequences=True,input_shape=(x_train.shape[-2:])))
lstm_model.add(Dense(20,activation='linear'))
lstm_model.add(LSTM(50))
lstm_model.add(Dropout(0.2))
lstm_model.add(Dense(units=horizon,activation='linear'))
lstm_model.add(Activation('linear'))
lstm_model.compile(loss='mean_squared_error', optimizer='adam')
lstm_model.summary()
history = lstm_model.fit(train_data,epochs=50,steps_per_epoch=100,validation_da
ta=val_data,validation_steps=50,verbose=1)
data_val = X_scaler.fit_transform(df[['Open', 'High', 'Low','Volume']].tail(10))
val_rescaled = data_val.reshape(1, data_val.shape[0], data_val.shape[1])
pred =lstm_model.predict(val_rescaled)
pred_Inverse = Y_scaler.inverse_transform(pred)
pred_Inverse
validate = (df[['Close']].tail(10))
timeseries_evaluation_metrics_func(validate['Close'],pred_Inverse[0])
```

Figure 66: Code Snippet for LSTM Implementations for Experiment 2

The difference between experiment 2 and experiment 1 is the feature selection part. Instead of using the "Pre_Close" as the input variable, experiment 2 uses "Open", "High", "Low", and "Volume" as the feature selection; the target variable is identical, which is "Close" indicates the Closing price that is companies as our objective by predicting the closing prices of metaverse based cryptocurrencies. Moreover, the data_val variable, the "Open", "High," "Low", and "Volume", need to be rescaled back to the original scale of the feature.

5.3.2 The Proposed GRUs Model

For the GRUs model, the experiment 2 flow is principally the same as the experiment 1, the code snippet, which are importing the library, importing the dataset of SAND metaverse cryptocurrencies, feature engineering, and data_preprocessing, is repeated the same with the experiment 1, as the library used, the dataset is the same. This can be observed in code snippet in Figure 67 below.

```
X_scaler = MinMaxScaler()
Y_scaler = MinMaxScaler()
X_data = X_scaler.fit_transform(df[[['Open', 'High', 'Low','Volume']])
Y_data = Y_scaler.fit_transform(df[['Close']])
look_back = 10
horizon = 10
train_split = int(len(df) * 0.8)
batch_size = 32
x_train, y_train = data_prep(X_data, Y_data, 0, train_split, look_back, horizon)
x_vali, y_vali = data_prep(X_data, Y_data, train_split, None, look_back, horizon)
train_data = tf.data.Dataset.from_tensor_slices((x_train, y_train))
train_data = train_data.batch(batch_size).repeat()
val_data = tf.data.Dataset.from_tensor_slices((x_vali, y_vali))
val_data = val_data.batch(batch_size).repeat()
tf.random.set_seed(7)
gru_model = Sequential()
gru_model.add(LSTM(100,return_sequences=True,input_shape=(x_train.shape[-2:])))
gru_model.add(Dense(20,activation='linear'))
gru_model.add(LSTM(50))
gru_model.add(Dropout(0.2))
gru_model.add(Dense(units=horizon,activation='linear'))
gru_model.add(Activation('linear'))
gru_model.compile(loss='mean_squared_error', optimizer='adam')
gru_model.summary()
history = gru_model.fit(train_data,epochs=50,steps_per_epoch=100,validation_da
ta=val_data,validation_steps=50,verbose=1)
data_val = X_scaler.fit_transform(df[['Open', 'High', 'Low','Volume']].tail(10))
val_rescaled = data_val.reshape(1, data_val.shape[0], data_val.shape[1])
pred =gru_model.predict(val_rescaled)
pred_Inverse = Y_scaler.inverse_transform(pred)
validate = (df[['Close']].tail(10))
timeseries_evaluation_metrics_func(validate['Close'],pred_Inverse[0])
```

Figure 67: Code Snippet for GRUs Model Implementations for experiment 2

The feature selection part is the difference between experiment 2 and experiment 1 for the GRUs model. Instead of using the 'Pre_Close' as the input variable, experiment 2 uses 'Open', 'High', 'Low', and 'Volume' as the feature selection; the target variable is identical, which are 'Close' indicates the closing price that is companies as our objective by predicting the closing prices of metaverse based cryptocurrencies. Moreover, the data_val variable, the 'Open', 'High,' 'Low', and 'Volume' need to be rescaled back to the original scale of the feature.

5.3.3 The Proposed CNN Model

For the CNN model, the experiment 2 flow is principally the same as the experiment 1, the code snippet, which is importing the library, importing the dataset of SAND metaverse cryptocurrencies, feature engineering, and data preprocessing, is repeated the same with the experiment 1, as the library used, the dataset is the same. It can be observed in the code snippet Figure 68 below.

```
X_scaler = MinMaxScaler()
Y_scaler = MinMaxScaler()
X_data = X_scaler.fit_transform(df[[['Open', 'High', 'Low','Volume']])
Y_data = Y_scaler.fit_transform(df[['Close']])
look_back = 10
horizon = 10
train_split = int(len(df) * 0.8)
batch_size = 32
x_train, y_train = data_prep(X_data, Y_data, 0, train_split, look_back, horizon)
x_vali, y_vali = data_prep(X_data, Y_data, train_split, None, look_back, horizon)
train_data = tf.data.Dataset.from_tensor_slices((x_train, y_train))
train_data = train_data.batch(batch_size).repeat()
val_data = tf.data.Dataset.from_tensor_slices((x_vali, y_vali))
val_data = val_data.batch(batch_size).repeat()
tf.random.set_seed(7)
cnn_model = Sequential()
```

```
cnn_model.add(Conv1D(filters=32, kernel_size=1, activation='relu', input_shape=(x_train.s
hape[-2:])))
cnn_model.add(Conv1D(filters=64, kernel_size=1))
cnn_model.add(MaxPooling1D(pool_size=(1)))
cnn_model.add(Conv1D(filters=128, kernel_size=1))
cnn_model.add(Flatten())
cnn_model.add(Dense(256,))
cnn_model.add(Dropout(0.2))
cnn_model.add(Dense(10))
cnn_model.compile(optimizer='adam', loss='mse')
cnn_model.summary()
history = gru_model.fit(train_data,epochs=50,steps_per_epoch=100,validation_da
ta=val_data,validation_steps=50,verbose=1)
data_val = X_scaler.fit_transform(df[['Open', 'High', 'Low','Volume']].tail(10))
val_rescaled = data_val.reshape(1, data_val.shape[0], data_val.shape[1])
pred =gru_model.predict(val_rescaled)
pred_Inverse = Y_scaler.inverse_transform(pred)
pred_Inverse
validate = (df[['Close']].tail(10))
timeseries_evaluation_metrics_func(validate['Close'],pred_Inverse[0])
```

Figure 68: Code Snippet for CNN Model Implementations for Experiment 2

The feature selection part is the difference between experiment 2 and experiment 1 for the CNN model. Instead of using the "Pre_Close" as the input variable, experiment 2 uses "Open", "High", "Low", and "Volume" as the feature selection; the target variable is identical, which is "Close" indicates the closing price that is companies as our objective by predicting the closing prices of metaverse based cryptocurrencies. Moreover, the data_val variable, the "Open", "High," "Low", and "Volume" need to be rescaled back to the original scale of the feature.

5.4 Experiment 3 Implementation

Experiment 3 is to investigate the performance of using the "Pre_Close", "H-L," "0-C", "SMA_7", "rsi_7", "EMA_7" feature to predict the next 10 days of SAND, SLP, and MANA, metaverse cryptocurrency closing price feature in terms of MAPE, MSE, MAPE, and RMSE by using three models which are LSTM, GRUs, and CNN. In 5.3 sections show the implementations of the proposed LSTM, GRUs, and CNN by using SAND data sets. Similarly, the whole steps in the 5.3 section need to repeat in this section using slp and mana datasets by altering the pd.read_csv(sand) function to pd.read_csv(slp) and pd.read_csv(mana) separately to predict both closing prices.

5.4.1 The Proposed LSTM Model

For the experiment 3 implementation of the LSTM model, we first need to repeat the steps stated in Figure 66. The steps included importing the library, dataset, feature engineering, data preprocessing, and model evaluation. The only alter things is the changing of the input feature in X_data from "Open", "High", "Low", "Volume" to "Pre_Close", "H-L", "O-C", "SMA_7", "rsi_7", "EMA_7".

5.4.2 The Proposed GRUs Model

For the experiment 3 implementation of the GRUs model, we first need to repeat the steps stated in Figure 67. The steps included importing the library, dataset, feature engineering, data preprocessing, and model evaluation. The only alter things is the changing of the input feature in X_data from "Open", "High", "Low", "Volume" to "Pre_Close", "H-L", "O-C", "SMA_7", "rsi_7", "EMA_7".

5.4.3 The Proposed CNN Model

For the experiment 3 implementation of the CNN model, we first need to repeat the steps stated in Figure 68. The steps included importing the library, dataset, feature engineering, data preprocessing, and model evaluation. The only alter things is the changing of the input feature in X_data from "Open", "High", "Low", "Volume" to "Pre_Close", "H-L", "O-C", "SMA_7", "rsi_7", "EMA_7".

5.5 Experiment 4 to 13 Implementations

Experiment 4 to 13 is to find the optimal features to predict the next 10 days of SAND, SLP, and MANA, metaverse cryptocurrency closing price feature in terms of MAPE, MSE, MAPE, and RMSE, by using three models, which are LSTM, GRUs, and CNN. In 5.3 sections show the implementations of the proposed LSTM, GRUs, and CNN by using SAND data sets. Similarly, the whole steps in the 5.3 section need to repeat I this section using slp and mana datasets by altering the pd.read_csv(sand) function to pd.read_csv(slp) and pd.read_csv(mana) separately to predict both closing prices.

5.5.1 The Proposed LSTM Model

For the experiment 4 to 13 implementation of the LSTM model, we first need to repeat the steps stated in Figure 49: Code Snippet for LSTM model implementations for experiment 2. The steps included importing the library, dataset, feature engineering, data preprocessing, and model evaluation. The only alter things is the changing of the input feature in X_data based on Table 14(Experiment 4), Table 17(Experiment 5), Table 20(Experiment 6), Table 23(Experiment 7), Table 26(Experiment 8), Table 29(Experiment 9), Table 32(Experiment 10), Table 35(Experiment 11), Table 38(Experiment 12) and Table 41(Experiment 13).

5.5.2 The Proposed GRUs Model

For the experiment 4 to 13 implementation of the GRUs model, we first need to repeat the steps stated in Figure 50: Code Snippet for GRUs model implementations for experiment 2. The steps included importing the library, dataset, feature engineering, data pre-processing, and model evaluation. The only alter things is the changing of the input feature in X_data based on Table 15 (Experiment 4), Table 18 (Experiment 5), Table 21(Experiment 6), Table 24(Experiment 7), Table 27(Experiment 8), Table 30 (Experiment 9), Table 33 (Experiment 10), Table 36 (Experiment 11), Table 39 (Experiment 12) and Table 41 (Experiment 13).

5.5.3 The Proposed CNN Model

For the experiment 4 to 13 implementation of the CNN model, we first need to repeat the steps stated in Figure 51, which is Code Snippet for CNN model implementations for experiment 2. The steps included importing the library, dataset, feature engineering, data preprocessing, and model evaluation. The only alter things is the changing of the input feature in X_data based on Table 16(Experiment 4), Table 19(Experiment 5), Table 22(Experiment 6), Table 25(Experiment 7), Table 28(Experiment 8), Table 31(Experiment 9), Table 34(Experiment 10), Table 37(Experiment 11), Table 40(Experiment 12) and Table 41(Experiment 13).

5.6 Experiment 14 to 20 Implementations

Experiment 4 to 13 is to find the optimal parameter set up of 4 feature selected (Previous Closing Price, Primary Technical Indicator, Advance Technical Indicator and Best Optimal Feature find throughout Experiments 4 to 13 to predict the next 10 days of closing price feature in terms of MAPE, MSE, MAPE, and RMSE, using LSTM, GRUs and CNN models. The hyperparameter tuning is completed throughout the experiment 14 to 20 using the different value of learning rate, batch size, complexity of the model architecture, regularization techniques such as L1 regularization, L2 regularization and dropout layer and epochs used for training and validation. In 5.3 sections show the implementations of the proposed LSTM, GRUs, and CNN by using SAND data sets. Similarly, the whole steps in the 5.3 section need to repeat using SLP and MANA datasets by altering the pd.read_csv(sand) function to pd.read_csv(slp) and pd.read_csv(mana) separately to predict both closing prices after completed the Experiment 20.

5.6.1 The Proposed LSTM Model

For the experiment 14 to 20 implementation of the LSTM model, we first need to repeat the steps stated in Figure 66: Code Snippet for LSTM Implementations for Experiment 2. The steps included importing the library, dataset, feature engineering, data pre-processing, and model evaluation. The only alter things is the changing of the hyperparameter set up in GRUs architecture based on table 45 (Experiment 14), table 49 (Experiment 15), table 54 (Experiment 16), table 59 (Experiment 17), table 63 (Experiment 18), table 68 (Experiment 19), table 72 (Experiment 20).

5.6.2 The Proposed GRUs Model

For the experiment 14 to 20 implementation of the GRUs model, we first need to repeat the steps stated in Figure 67: Code Snippet for GRUs Model Implementations for experiment 2. The steps included importing the library, dataset, feature engineering, data pre-processing, and model evaluation. The only alter things is the changing of the hyperparameter set up in GRUs architecture based on Table 46(Experiment 14), Table 50(Experiment 15), Table 55(Experiment 16), Table 60(Experiment 17), Table 64(Experiment 18), Table 69(Experiment 19), Table 73(Experiment 20).

5.6.3 The Proposed CNN Model

For the experiment 4 to 13 implementation of the CNN model, we first need to repeat the steps stated in Figure 68: Code Snippet for CNN Model Implementations for Experiment 2 The steps included importing the library, dataset, feature engineering, data preprocessing, and model evaluation. The only alter things is the changing of the hyperparameter set up in CNN architecture based on Table 47(Experiment 14), Table 51(Experiment 15), Table 56(Experiment 16), Table 61(Experiment 17), Table 65(Experiment 18), Table 70(Experiment 19), Table 74(Experiment 20).

5.7 Conclusion

In conclusion, chapter 5 describes the Python code implementation of experiment 1 to 20 by using Google Colab as the platform for SAND, SLP and MANA metaverse cryptocurrencies 10 days future price prediction by using proposed LSTM, GRUs and CNN model. The flow of code implementation is explained explicitly throughout the chapter, including data and feature engineering used in experiment 1 to 20, model implementation of the proposed LSTM, GRUs and CNN model and the model evaluation metrics.

CHAPTER 6

EXPERIMENTS RESULTS

6.1 Feature Selection Experiments (1-13)

This section discusses the results of experiments 1 to 13 to three metaverse cryptocurrencies, SAND, SLP and MANA, in the metrics of MSE, MAE, RMSE, and MAPE. SAND will be the primary cryptocurrencies, and the SLP and MANA will be the secondary cryptocurrencies used for the test cases. With a comparative analysis of the value of the MAPE in these experiments, the best model and the feature used can be determined for these three metaverse cryptocurrencies. The first three experiments will evaluate the predictivity of the Previous Closing Price(PC), the Primary Technical Indicator (PTI) features and advanced technical indicators which are used "H-L", "0-C", "SMA_7", "rsi_7", and "EMA_7" as the input features by using the SAND cryptocurrency. These three experiments will also be evaluated using SLP and MANA for testing. Finally, the exhaustive search of feature combinations is done by the remained experiments, which are experiments 4 to 13, for the best optimal feature(BOF) that allows one to build accurate models.

6.1.1 Experiment 1 (Previous Close)

The experiment 1 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in the after sections implicitly. In addition, experiment 1 uses "Prev_Close" as an input feature to predict the next ten days of closing price, labelled as "Close" target features.

6.1.1.1 LSTM

Table 5: Results of LSTM model in Experiment 1

Cryptocurrencies	Metrics			
	MSE	MAE	RMSE	MAPE
SAND	5.5697	2.3422	2.3600	82.9270
SLP	0.0043	0.0627	0.0657	351.9648
MANA	0.8891	0.9393	0.9429	42.9516

Table 5 above summarizes the results LSTM model in Experiment 1. In experiment 1, by using LSTM as the predictive model, MAPE has been recorded as 82.9270 to predict the closing price of the SAND. Therefore, the MAPE of the LSTM is recorded as 82.9270 means the average of the absolute percentage errors of forecasts is 82.9270 %. In the other test case, using other cryptocurrencies, SLP and MANA, record as means absolute percentage errors of forecasts of 351.9648% and 42.9516%.

6.1.1.2 GRUs

Table 6: Results of GRUs model in Experiment 1

Cryptocurrencies	Metrics			
	MSE	MAE	RMSE	MAPE
SAND	0.8260	0.7819	0.9088	27.9069
SLP	0.0019	0.0419	0.0441	236.1956
MANA	1.0526	1.0220	1.0260	46.7369

Table 6 above summarizes the results of GRUs model in Experiment 1. In experiment 1, by using GRUs as the predictive model, MAPE has been recorded as 27.9069 to predict the closing price of the SAND. Therefore, the MAPE of the GRUs is recorded as 27.9069 means the average of the absolute percentage errors of forecasts is 27.9069%. Another test case using other cryptocurrencies, SLP and MANA, produced mean absolute percentage errors of forecasts of 236.1946% and 46.7318%.

6.1.1.3 CNN

Table 7: Results of CNN model in Experiment 1

Cryptocurrencies	Metrics			
	MSE	MAE	RMSE	MAPE
SAND	1.3917	0.9503	1.1797	33.6532
SLP	0.0006	0.0219	0.0246	122.9813
MANA	0.0690	0.2517	0.2628	11.4615

Table 7 above summarizes the results of CNN model in Experiment 1. In experiment 1, by using CNN as the predictive model, MAPE has been recorded as 33.6532 to predict the closing price of the SAND. Therefore, the MAPE of the CNN is recorded as 33.6532 means the average of the absolute percentage errors of forecasts is 33.6532 %. Another test case using other cryptocurrencies, SLP and MANA, produced mean absolute percentage errors of forecasts of 122.9813 % and 11.4615 %.

6.1.2 Experiment 2 (Primary Technical Indicator)

The experiment 2 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in this section implicitly. Experiment 2 uses "Open", "High", "Low", and "Volume" as input features to predict the next ten days of closing price, labelled as "Close" target features.

6.1.2.1 LSTM

Table 8: Results of LSTM model in Experiment 2

Cryptocurrencies	Metrics			
	MSE	MAE	RMSE	MAPE
SAND	10.3318	3.1820	3.2143	112.8267
SLP	0.0013	0.0353	0.0354	200.9148
MANA	0.1563	0.3839	0.3953	17.5064

Table 8 above summarizes the results of LSTM model in Experiment 2. In experiment 2, using LSTM as the predictive model, MAPE has been recorded as 112.8267 to predict the closing price of the SAND. Therefore, the MAPE of the LSTM is recorded as 112.8267 means the average of the absolute percentage errors of forecasts is 112.8267%. In the other test case, using other cryptocurrencies, SLP and MANA, record as means absolute percentage errors of forecasts of 200.9148% and 17.5064%.

6.1.2.2 GRUs

Table 9: Results of GRUs model in Experiment 2

Cryptocurrencies	Metrics			
	MSE	MAE	RMSE	MAPE
SAND	0.8384	0.7150	0.9156	25.2571
SLP	0.0278	0.1628	0.1668	933.8402
MANA	2.5632	1.5957	1.6010	73.0209

Table 9 above summarizes the results of GRUs model in Experiment 2. In experiment 2, by using GRUs as the predictive model, MAPE has been recorded as 25.2571 to predict the closing price of the SAND. Therefore, the MAPE of the LSTM is recorded as 25.2571 means the average of the absolute percentage errors of forecasts is 25.2571 %. In the other test case, using other cryptocurrencies, SLP and MANA, record as means absolute percentage errors of forecasts of 933.8402% and 73.0209%.

6.1.2.3 CNN

Table 10: Results of CNN model in Experiment 2

Cryptocurrencies	Metrics			
	MSE	MAE	RMSE	MAPE
SAND	6.6546	2.1983	2.5796	77.3749
SLP	0.0008	0.0243	0.0291	77.3749
MANA	1.6406	1.2563	1.2809	57.4322

Table 10 above summarizes the results of CNN model in Experiment 2. In experiment 2, by using CNN as the predictive model, MAPE has been recorded as 77.3749 to predict the closing price of the SAND. Therefore, the MAPE of the CNN is recorded as 77.3749 means the average of the absolute percentage errors of forecasts is 77.3749 %. In the other test case, using other cryptocurrencies, SLP and MANA, record as means absolute percentage errors of forecasts of 77.3749 % and 57.4322 %.

6.1.3 Experiment 3 (Advanced Technical Indicator)

The experiment 3 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in this section implicitly. Experiment 3 uses "H-L", "0-C", "SMA_7", "rsi_7" and "EMA_7" as input features to predict the next ten days of closing price, which are labelled as "Close" target features.

6.1.3.1 LSTM

Table 11: Results of LSTM model in Experiment 3

Cryptocurrencies	Metrics			
	MSE	MAE	RMSE	MAPE
SAND	1.0109	3.6440	3.6498	128.9782
SLP	0.0072	0.0845	0.0851	478.2201
MANA	2.1293	1.4565	1.4592	66.6537

Table 11 above summarizes the results of LSTM model in Experiment 3. In experiment 3, by using LSTM as the predictive model, MAPE has been recorded as 128.9782 to predict the closing price of the SAND. Therefore, the MAPE of the LSTM is recorded as 128.9782 means the average of the absolute percentage errors of forecasts is 128.9782 %. In the other test case, using other cryptocurrencies, SLP and MANA, record as means absolute percentage errors of forecasts of 478.2201% and 66.644%.

6.1.3.2 GRUs

Table 12: Results of GRUs model in Experiment 3

Cryptocurrencies	Metrics			
	MSE	MAE	RMSE	MAPE
SAND	1.9047	0.8952	1.0054	31.2915
SLP	0.0042	0.0631	0.0645	356.3695
MANA	4.1786	2.0419	2.0442	93.4819

Table 12 above summarizes the results of GRUs model in Experiment 3. In experiment 3, by using GRUs as the predictive model, MAPE has been recorded as 31.2915 to predict the closing price of the SAND. Therefore, the MAPE of the GRUs is recorded as 31.2915 means the average of the absolute percentage errors of forecasts is 31.2915 %. In the other test case, using other cryptocurrencies, SLP and MANA, record as means absolute percentage errors of forecasts of 356.3695% and 93.4819%.

6.1.3.3 CNN

Table 13: Results of CNN model in Experiment 3

Cryptocurrencies	Metrics			
	MSE	MAE	RMSE	MAPE
SAND	7.2024	2.4872	2.6837	88.2758
SLP	0.0180	0.1211	0.1342	674.6060
MANA	0.5524	0.7069	0.7433	32.2474

Table 13 above summarizes the results of CNN model in Experiment 3. In experiment 3, by using CNN as the predictive model, MAPE has been recorded as 88.2758 to predict the closing price of the SAND. Therefore, the MAPE of the LSTM is recorded as 88.2758 means the average of the absolute percentage errors of forecasts is 88.2758 %. In the other test case, using other cryptocurrencies, SLP and MANA, record as means absolute percentage errors of forecasts of 674.6060 % and 32.2474%.

6.1.4 Experiment 4 (1 features)

The results of experiment 4 have been tabulated using MSE, MAE, RMSE, and MAPE metrics and are implicitly discussed in this section. Experiment 4 employs one dependent feature as an input feature to predict the next ten days' closing price, designated as the "Close" target feature. Experiment 4 evaluates the predictability of 10 dependent features, including "Open", "High", "Low", "Volume", "Pre_Close", "H-L", "0-C", "SMA_7", "rsi_7", and "EMA_7", by feeding each feature individually into LSTM, GRU, and CNN models. SAND will be the primary metaverse cryptocurrency used to evaluate the best predictivity of the feature and the hyperparameter tuning process to select the model with the best combination of features. The selection feature will be evaluated once more by implementing the SLP and MANA datasets.

6.1.4.1 LSTM

Table 14: Results of SAND Prediction by LSTM model in Experiment 4

Feature	Metrics			
	MSE	MAE	RMSE	MAPE
Open	4.4785	2.0370	2.1162	72.2716
High	0.9602	0.8349	0.9799	29.2304
Low	5.7914	2.3973	2.4065	84.8564
Volume	**0.2286**	**0.3776**	**0.4782**	**13.3211**
Pre_Close	5.5697	2.3422	2.3600	82.9270
H-L	5.4599	2.3257	2.3366	82.1717
0-C	13.3403	3.6425	3.6524	128.9544
SMA_7	5.6923	2.3396	2.3858	82.9237
rsi_7	3.0757	1.7458	1.7538	61.5621
EMA_7	6.4878	2.5369	2.5471	89.7938

Note: Bolded values indicate the best results.

Table 14 above summarizes the results of SAND Prediction by LSTM model in Experiment 4. In experiment 4, applying LSTM as the predictive model, "Volume" outperformed the other features in predicting the closing price of the SAND metaverse cryptocurrency, as the four metrics values recorded the lowest value compared to the other nine dependent features. Furthermore, the fact that the MAPE of the "Volume" feature is 13.3211 indicates that the average absolute percentage error of forecasts is 13.3211%.

6.1.4.2 GRUs

Table 15: Results of SAND Prediction by GRUs model in Experiment 4

Feature	Metrics			
	MSE	MAE	RMSE	MAPE
Open	0.8907	0.8147	0.9438	29.0328
High	6.3392	2.5043	2.5178	88.4685
Low	2.5246	1.4510	1.5889	51.6319
Volume	**0.1503**	**0.2846**	**0.3877**	**10.0686**
Pre_Close	0.8259	0.7819	0.9088	27.9049
H-L	1.8273	1.3126	1.3518	46.2911
0-C	15.6609	3.9390	3.9574	139.3973
SMA_7	13.7488	3.6976	3.7079	130.5644
rsi_7	3.7869	1.9374	1.9460	68.3178
EMA_7	19.0914	4.3580	4.3694	153.843

Note: Bolded values indicate the best results.

Table 15 above summarizes the results of SAND Prediction by GRUs model in Experiment 4. In experiment 4 utilising GRUs as the predictive model, "Volume" outperformed the other features in predicting the closing price of the SAND metaverse cryptocurrency since the four metrics values recorded the lowest value compared to the other nine dependent features. The MAPE of "Volume" features is 10.0686 shows that the average absolute percentage error of forecasts is 10.0686 %.

6.1.4.3 CNN

Table 16: Results of SAND Prediction by CNN model in Experiment 4

Feature	Metrics			
	MSE	MAE	RMSE	MAPE
Open	0.9802	0.7620	0.9900	27.0086
High	2.4348	1.3270	1.5604	46.4680
Low	2.0750	1.1681	1.4405	41.6225
Volume	**0.2294**	**0.4138**	**0.4790**	**14.5994**
Pre_Close	1.3917	0.9503	1.1797	33.6532
H-L	1.2134	0.9027	1.1015	32.0230
0-C	66.6429	8.0729	8.1635	285.8198
SMA_7	2.9302	1.6021	1.7118	56.3646
rsi_7	2.6898	1.6337	1.6400	57.6370
EMA_7	1.2483	1.0543	1.1173	37.0530

Note: Bolded values indicate the best results.

Table 16 above summarizes the results of SAND Prediction GRUs model in Experiment 4. In experiment 4, using the CNN as the predictive model, 'Volume' outperformed the other features in predicting the closing price of the SAND metaverse cryptocurrency, as the four metrics values recorded the lowest value when compared to the other nine dependent features. The MAPE of "Volume" features is 14.5994, indicating that the average absolute percentage error of forecasts is 14.5994 %.

6.1.5 Experiment 5 (2 features)

The results of experiment 5 have been tabulated using MSE, MAE, RMSE, and MAPE metrics and are implicitly discussed in this section. Experiment 5 uses two dependent features as input features to predict the closing price for the next ten trading days, which are labelled as "Close" target features. Experiment 5 evaluates the predictability of two features by testing every possible combination of the two features using the grid search method. The nCr formula is used to determine the count of the many ways in which r things may be picked from n different items when the order is not considered.

$$nCr = \frac{n!}{r!\,(n-r)!}$$

Where, n is the count of the many ways, r is the number of things to be chosen out of n items. According to the combination formula, there are 45 test cases for each of the LSTM, GRU, and CNN models. SAND will be the primary cryptocurrency used to assess the predictability of two-feature combinations. For the subsequent experiments, the top five two-feature combinations with the lowest MSE, MAE, MSE, and MAPE values will be chosen for feature addition.

6.1.5.1 LSTM

Table 17: Results of SAND Prediction by LSTM model in Experiment 5

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High	13.5691	3.6647	3.6836	129.7726
Open, Low	4.2577	1.9131	2.0634	67.9975
Open, Volume	9.0627	3.0075	106.3054	106.3054

Open, Pre_Close	**3.2855**	**1.6371**	**1.8126**	**58.1844**
Open, H-L	7.6588	2.7641	2.7675	97.7832
Open, 0-C	8.1037	2.8453	2.8467	100.6323
Open, SMA_7	5.4013	2.1585	2.3241	76.6842
Open, rsi_7	10.5417	3.1476	3.2468	111.6636
Open, EMA_7	6.6198	2.5233	2.5729	89.4645
High, Low	**2.9994**	**1.3343**	**1.7319**	**47.5484**
High, Volume	9.8285	3.1212	3.135	110.5297
High, Pre_Close	**2.1969**	**1.1754**	**1.4822**	**41.5457**
High, H-L	4.4725	1.9375	2.1148	68.8877
High, 0-C	**2.4611**	**1.3849**	**1.5688**	**49.2561**
High, SMA_7	**4.4059**	**1.6931**	**2.099**	**60.3516**
High_rsi_7	11.6465	3.0888	3.4127	109.9024
High_EMA_7	4.6113	1.7281	2.1474	61.6944
Low, Volume	7.5568	2.7478	2.749	97.1794
Low, Pre_Close	7.6058	2.7404	2.7579	97.0296
Low, H-L	6.0829	2.463	2.4664	87.1281
Low, 0-C	7.1281	2.6689	2.6699	94.3576
Low, SMA_7	5.8322	2.2775	2.415	80.8931
Low, rsi_7	8.5118	2.8757	2.9175	101.931
Low, EMA_7	6.7399	2.568	2.5961	90.9815
Volume, Pre_Close	8.546	2.917	2.9234	103.1058
Volume, H-L	3.556	1.8808	1.8857	66.4701
Volume, 0-C	3.3457	1.8091	1.8291	63.8957
Volume, SMA_7	11.6509	3.385	3.4133	119.535
Volume, rsi_7	2.258	2.258	2.2624	79.7844
Volume, EMA_7	13.8183	3.6799	3.7173	129.9948
Pre_Close, H-L	7.0334	2.6483	2.6521	93.67
Pre_Close, 0-C	7.2584	2.6931	2.6941	95.2185
Pre_Close, SMA_7	5.7864	2.3389	2.4055	82.9639
Pre_Close, rsi_7	7.6984	2.7392	2.7746	97.0752
Pre_Close, EMA_7	7.0422	2.6168	2.6537	92.7373
H-L, 0-C	4.1951	2.0398	2.0482	72.0721
H-L, SMA_7	10.456	3.2152	3.2336	113.563
H-L, rsi_7	4.7836	2.1843	2.1871	77.2525
H-L, EMA_7	10.8717	3.2796	3.2972	115.8618
0-C, SMA_7	8.8638	2.976	2.9772	105.2492
0-C, rsi_7	16.0182	3.9663	4.0023	140.2246
0-C, EMA_7	10.0599	3.1699	3.1717	112.0945
SMA_7, rsi_7	12.9003	3.4716	3.5917	123.1818
SMA_7, EMA_7	6.7859	2.5241	2.605	89.5556
rsi_7, EMA_7	9.024	2.8156	3.004	100.0331

Note: Bolded values indicate the 5 best results(2 features).The 5 best results features will be selected for the next LSTM experiments by adding up 1 feature.

Table 17 above summarizes the results of SAND Prediction by LSTM model in Experiment 5. The top five two-feature combinations in experiment 5 with LSTM as the predictive model are "Open, Pre_Close", "High, Low", "High, Pre_Close", "High, 0-C," and "High, SMA_7". These combinations have the top five lowest MAPE values, recorded as 58.1844, 47.5484, 41.5457, 49.2561, and 60.3516. Furthermore, out of all these features, the "High,0-C" has the highest predictive power, with an average absolute percentage error of forecasts of just 41.5457%.

6.1.5.2 GRUs

Table 18: Results of SAND Prediction by GRUs model in Experiment 5

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High	2.3878	1.4608	1.5453	51.3783
Open, Low	2.0903	1.3342	1.4458	47.4902
Open, Volume	0.3337	0.4841	0.5777	17.0689
Open, Pre_Close	1.5357	1.0732	1.2392	38.2407
Open, H-L	0.1965	0.3710	0.4433	13.0000
Open, 0-C	1.8173	1.1385	1.3481	40.5674
Open, SMA_7	0.7923	0.7582	0.8901	26.5568
Open, rsi_7	5.7828	2.4003	2.4047	84.9034
Open, EMA_7	1.6942	1.2039	1.3016	42.2926
High, Low	1.7325	1.2554	1.3162	44.1464
High, Volume	5.8786	2.3855	2.4246	84.1460
High, Pre_Close	1.6649	1.2111	1.2903	42.5658
High, H-L	0.7457	0.7844	0.8635	27.5746
High, 0-C	2.6569	2.6569	2.6720	93.6865
High, SMA_7	5.7046	2.3480	2.3884	82.7938
High_rsi_7	2.1452	1.4229	1.4646	50.0515
High_EMA_7	6.3842	2.5026	2.5267	88.3081
Low, Volume	1.3232	1.0195	1.1503	36.2740
Low, Pre_Close	1.6821	1.2175	1.2969	43.2743
Low, H-L	2.4874	1.5344	1.5771	54.4026
Low, 0-C	4.2301	1.9965	2.0567	70.7926
Low, SMA_7	1.2521	0.8833	1.1190	31.3635
Low, rsi_7	9.0847	2.9164	3.0141	103.5365
Low, EMA_7	0.8162	0.7838	0.9035	27.4539
Volume, Pre_Close	2.4282	1.3871	1.5583	49.3623
Volume, H-L	1.8618	1.3210	1.3645	46.5786
Volume, 0-C	0.2242	0.4198	0.4735	14.8920
Volume, SMA_7	5.2341	2.2462	2.2878	79.1775
Volume, rsi_7	0.2771	0.4937	0.5264	17.4799
Volume, EMA_7	7.9017	2.7882	2.8110	98.4010
Pre_Close, H-L	1.8945	1.2955	1.3764	46.0155
Pre_Close, 0-C	1.8427	1.2142	1.3575	43.2134
Pre_Close, SMA_7	0.5475	0.6584	0.7399	23.0660
Pre_Close, rsi_7	7.6502	2.6200	2.7659	93.1320
Pre_Close, EMA_7	1.9155	1.2725	1.3840	44.6739
H-L, 0-C	3.4836	1.8193	1.8664	64.1409
H-L, SMA_7	3.4322	1.7372	1.8526	61.0872
H-L, rsi_7	10.2029	3.1473	3.1942	111.1409
H-L, EMA_7	14.8362	3.8439	3.8518	135.7933
0-C, SMA_7	2.0373	1.2012	1.4273	42.2601
0-C, rsi_7	6.7235	2.5538	2.5930	90.5578
0-C, EMA_7	2.7329	1.4888	1.6531	52.3914
SMA_7, rsi_7	0.5069	0.5851	0.7119	20.4004

SMA_7, EMA_7	10.3630	3.1861	3.2192	112.5259
rsi_7, EMA_7	2.8614	1.6839	1.6916	59.3683

Note: Bolded values indicate the 5 best results(2 features).The 5 best results features will be selected for the next GRUs experiments by adding up 1 feature.

Table 18 above summarizes the results of SAND Prediction by GRUs model in Experiment 5. The top five two-feature combinations in experiment 5 with GRUs as the predictive model are "Open, Volume", "Open, H-L", "Volume, 0-C", "Volume, rsi_7", and "SMA_7, rsi_7" with respective MAPE values of 17.0689, 13, 14.892, 17.4799, and 20.4004. The feature with the highest predictive power among these is the "High,0-C," with an average absolute percentage error of forecasts of 13%.

6.1.5.3 CNN

Table 19: Results of SAND Prediction by CNN model in Experiment 5

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High	2.5573	1.3996	1.5991	49.0803
Open, Low	31.5784	5.3241	5.6195	188.3417
Open, Volume	0.7320	0.5835	0.8555	20.7014
Open, Pre_Close	2.1037	1.0813	1.4504	38.4246
Open, H-L	1.8302	1.2812	1.3529	45.3086
Open, 0-C	1.3529	1.5871	1.6581	56.4151
Open, SMA_7	0.9488	0.8063	0.9741	28.1130
Open, rsi_7	6.8883	2.5983	2.6246	92.0878
Open, EMA_7	**0.1777**	**0.3682**	**0.4215**	**13.1515**
High, Low	1.9329	1.2622	1.3903	44.5873
High, Volume	1.5567	1.0313	1.2477	36.7840
High, Pre_Close	1.7752	1.1508	1.3324	40.3254
High, H-L	2.4404	1.3033	1.5622	46.1845
High, 0-C	**0.3140**	**0.4661**	**0.5603**	**16.5978**
High, SMA_7	**0.1869**	**0.3794**	**0.4323**	**13.3742**
High_rsi_7	0.3236	0.4799	0.5689	16.9045
High_EMA_7	1.2877	0.9774	1.1348	34.3029
Low, Volume	2.0208	1.1216	1.4215	39.5980
Low, Pre_Close	1.1890	0.9142	1.0904	31.9694
Low, H-L	4.4798	1.9659	2.1165	69.8367
Low, 0-C	**0.1148**	**0.3031**	**0.3388**	**10.8099**
Low, SMA_7	1.5535	1.2049	1.2464	42.6524
Low, rsi_7	10.1181	3.1114	3.1809	110.3347
Low, EMA_7	8.6399	2.8681	2.9394	101.1444
Volume, Pre_Close	2.8687	1.2352	1.6937	43.3643
Volume, H-L	4.8241	1.9486	2.1964	68.9916
Volume, 0-C	93.6062	9.3873	9.6750	333.0903
Volume, SMA_7	0.8142	0.8483	0.9023	29.9148
Volume, rsi_7	3.1464	1.6349	1.7738	57.9491
Volume, EMA_7	1.0553	0.9652	1.0273	33.9453
Pre_Close, H-L	0.9150	0.7134	0.9566	25.5513

Pre_Close, 0-C	1.3077	1.0868	1.1436	38.5519
Pre_Close, SMA_7	0.8584	0.7619	0.9265	26.9773
Pre_Close, rsi_7	2.8146	1.6607	1.6777	58.7721
Pre_Close, EMA_7	2.1787	1.2682	1.4761	44.6887
H-L, 0-C	13.1276	3.4666	3.6232	122.5434
H-L, SMA_7	7.1520	2.4109	2.6743	85.9227
H-L, rsi_7	9.4035	2.6210	3.0665	92.3516
H-L, EMA_7	0.9843	0.6977	0.9921	24.4564
0-C, SMA_7	0.9244	0.8263	0.9614	29.0398
0-C, rsi_7	71.4199	8.3576	8.4510	295.8712
0-C, EMA_7	1.0647	0.9130	1.0318	32.0697
SMA_7, rsi_7	1.0647	0.9130	1.0318	26.3935
SMA_7, EMA_7	2.2193	1.2896	1.4897	45.1271
rsi_7, EMA_7	**0.0617**	**0.1974**	**0.2483**	**6.9224**

Note: Bolded values indicate the 5 best results(2 features).The 5 best results features will be selected for the next CNN experiments by adding up 1 feature.

Table 19 above summarizes the results of SAND Prediction by CNN model in Experiment 5. The top five two-feature combinations in experiment 5 using CNN as the predictive model are "Open, EMA_7", "High, 0-C", "High, SMA_7", "Low, 0-C", and "rsi_7, EMA_7"; these values are 13.1515, 16.5978, 13.3742, 10.8099, and 6.9224. The lowest average of the absolute percentage errors of forecasts is 6.9224%, making the "rsi_7, EMA_7" feature among these have the highest predictive power.

6.1.6 Experiment 6 (3 features)

The experiment 6 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in this section implicitly. Experiment 6 uses three dependent features as input to predict the next ten days of closing price, labelled as "Close" target features. Experiment 6 evaluates the predictability of three-feature combinations by adding one more feature individually to the five selective two-feature combinations with the lowest MAPE value in experiment 5. In experiment 6, there are 34, 36 and 40 test cases for each model: LSTM, GRUs and CNN. SAND will be the primary metaverse cryptocurrencies used to evaluate the predictivity of the three-feature combinations. Five selective three-feature combinations with the lowest MSE, MAE, MSE and MAPE values will be selected for the following experiments.

6.1.6.1 LSTM

Table 20: Results of SAND Prediction by LSTM model in Experiment 6

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Pre_Close	3.6683	1.6987	1.9153	59.7629
High, Low, Pre_Close	**2.7188**	**1.3155**	**1.6489**	**46.5641**
High, Volume, Pre_Close	8.8171	2.9247	2.9694	103.6552
High, Pre_Close, H-L	12.1915	3.4865	3.4916	123.3639
High, Pre_Close,0-C	5.8933	2.3459	2.4276	83.2442
High, Pre_Close, SMA_7	3.7884	1.8794	1.9464	66.6922
High, Pre_Close, rsi_7	6.4378	2.0463	2.5373	73.0555
High, Pre_Close, EMA_7	**3.9049**	**1.6595**	**1.9761**	**58.6527**
Open, High, Low	4.1699	1.7731	2.0420	62.3850
High, Low, Volume	9.0696	3.0000	3.0116	106.2827
High, Low, H-L	11.9703	3.4563	3.4598	122.2401
High, Low, 0-C	4.2885	1.6862	2.0709	60.1805
High, Low, SMA_7	**4.4056**	**1.6592**	**2.0990**	**58.7611**
High, Low, rsi_7	6.8537	2.0927	2.6180	74.6793
High, Low, EMA_7	4.8768	1.7575	2.2084	62.7491
Open, High, 0-C	7.7611	2.7566	2.7859	97.7088
High, Volume, 0-C	7.3190	2.6379	2.7054	93.6393
High, H-L, 0-C	6.3597	2.4721	2.5219	87.7064
High, 0-C, SMA_7	6.4174	2.4769	2.5332	87.8133
High, 0-C, rsi_7	8.0814	2.7592	2.8428	97.9040
High, 0-C, EMA_7	6.7101	2.5277	2.5904	89.6225
Open, High, Pre_Close	3.6683	1.6987	1.9153	59.7629
Open, Low, Pre_Close	**2.8605**	**1.4277**	**1.6913**	**50.8485**
Open, Volume, Pre_Close	8.7817	2.9478	2.9634	104.3728
Open, Pre_Close, H-L	9.9307	3.1415	3.1513	111.0021
Open, Pre_Close, 0-C	6.6367	2.4669	2.5762	87.5728
Open, Pre_Close, SMA_7	4.7033	1.8301	2.1687	65.2226
Open, Pre_Close, rsi_7	9.3368	2.7406	3.0556	97.5376
Open, Pre_Close, EMA_7	6.0449	2.3252	2.4586	82.5753
Open, High, SMA_7	**2.0977**	**1.1534**	**1.4483**	**40.7961**
High, Volume, SMA_7	9.7442	3.0986	3.1216	109.7474
High, H-L, SMA_7	10.7395	3.2746	3.2771	115.7991
High, SMA_7, rsi_7	8.6137	2.3949	2.9349	85.4363
High, SMA_7, EMA_7	8.0874	2.4273	2.8438	85.7211

Note: Bolded values indicate the 5 best results(3 features).The 5 best results features will be selected for the next LSTM experiments by adding up 1 feature.

Table 20 above summarizes the results of SAND Prediction by LSTM model in Experiment 6. In experiment 6 employing LSTM as the predictive model, "High, Low, Pre_Close", "High, Pre_Close, EMA_7", "High, Low, SMA_7", "Open, Low, Pre_Close" and "Open, High, SMA_7" are the top five three-feature combinations with the lowest MAPE values, which are 46.5641, 58.6527, 58.7611, 50.8485, and 40.7961, respectively And among these features, "Open, High, SMA 7" has the strongest predictive power, as the average absolute percentage error of forecasts is the smallest at 40.7961%.

6.1.6.2 GRUs

Table 21: Results of SAND Prediction by GRUs model in Experiment 6

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Volume	1.1001	0.9346	1.0489	32.7977
Open, Low, Volume	1.4139	1.0872	1.1891	38.6639
Open, Volume, Pre_Close	3.1935	1.6599	1.7870	59.0217
Open, Volume, H-L	**0.3885**	**0.4050**	**0.6233**	**14.3824**
Open, Volume, 0-C	3.3439	1.6514	1.8286	58.8217
Open, Volume, SMA_7	3.6068	1.6137	1.8992	57.5110
Open, Volume, rsi_7	4.3877	1.7641	2.0947	62.9898
Open, Volume, EMA_7	2.2565	1.2546	1.5022	44.7437
Open, High, H-L	2.0272	1.3732	1.4238	48.4008
Open, Low, H-L	0.6640	0.6374	0.8149	22.7762
Open, Pre_Close, H-L	**0.8631**	**0.8076**	**0.9291**	**28.7439**
Open, H-L, 0-C	2.1996	1.4208	1.4831	50.4283
Open, H-L, SMA_7	1.4296	1.0863	1.1957	38.2313
Open, H-L, rsi_7	**0.4583**	**0.5694**	**0.6770**	**19.9205**
Open, H-L, EMA_7	1.8909	1.2867	1.3751	45.3436
High, Volume, 0-C	1.9395	1.3052	1.3927	46.4049
Low, Volume, 0-C	3.6020	1.8401	1.8979	65.2800
Volume, Pre_Close, 0-C	2.7141	1.5508	1.6474	55.0799
Volume, 0-C, SMA_7	0.9886	0.7846	0.9943	27.7106
Volume, 0-C, rsi_7	2.5210	1.5708	1.5878	55.4928
Volume, 0-C, EMA_7	**0.7932**	**0.8141**	**0.8906**	**28.5995**
High, Volume, rsi_7	2.6864	1.6187	1.6390	57.0580
Low, Volume, rsi_7	1.0384	0.8410	1.0190	30.0512
Volume, Pre_Close, rsi_7	**0.7320**	**0.7338**	**0.8555**	**26.1879**
Volume, SMA_7, rsi_7	1.1103	0.8912	1.0537	31.2355
Volume, rsi_7, EMA_7	1.6491	1.2385	1.2842	43.5818
High, SMA_7, rsi_7	3.8261	1.6330	1.9560	57.1656
Low, SMA_7, rsi_7	4.1339	1.6869	2.0332	60.0052
Pre_Close, SMA_7, rsi_7	3.2737	1.5686	1.8093	55.2734
SMA_7, rsi_7, EMA_7	2.0675	1.3313	1.4379	46.6874
Open, Volume, 0-C	3.3488	1.7199	1.8300	61.1128
Volume, H-L, 0-C	2.0945	1.4052	1.4472	49.5570
Open, Volume, rsi_7	1.5390	1.0128	1.2406	36.0968
Volume, H-L, rsi_7	3.0244	1.7197	1.7391	60.7194
Open, SMA_7, rsi_7	3.1027	1.5097	1.7615	53.2118
H-L, SMA_7, rsi_7	1.4687	1.0096	1.2119	35.2955
0-C, SMA_7, rsi_7	5.2940	2.2055	2.3009	78.5126

Note: Bolded values indicate the 5 best results(3 features).The 5 best results features will be selected for the next GRUs experiments by adding up 1 feature.

Table 21 above summarizes the results of SAND Prediction by GRUs model in Experiment 6. In experiment 6 utilising GRUs as the predictive model, "Open, Volume, H-L", "Open, Pre_Close, H-L", "Open, H-L, rsi_7", "Volume, 0-C, EMA_7", and "Volume, Pre_Close, rsi_7" are the top five three-feature combinations with the lowest MAPE values, which are 14.3824, 28.7439, 19.9205, 28.5995. And among these features, "Open, H-L, and rsi_7" have the greatest predictive power, as the average absolute percentage error of forecasts is 19.9205%.

6.1.6.3 CNN

Table 22: Results of SAND Prediction by CNN model in Experiment 6

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, rsi7, EMA7	0.6077	0.6242	0.7795	21.8666
High, rsi_7, EMA_7	**0.1333**	**0.3336**	**0.3651**	**11.8246**
Low, rsi_7, EMA_7	1.2019	0.9347	1.0963	33.0119
Volume, rsi_7, EMA_7	3.5454	1.2631	1.8829	44.8367
Pre, Close, rsi_7, EMA_7	0.6520	0.7122	0.8075	25.0658
H-L, rsi_7, EMA_7	0.6464	0.6129	0.8040	21.5728
0-C, rsi_7, EMA_7	0.5307	0.6365	0.7285	22.4522
SMA_7, rsi_7, EMA_7	**0.3970**	**0.5332**	**0.6301**	**19.0058**
Open, Low, 0-C	0.8815	0.8350	0.9389	29.4514
High, Low, 0-C	2.4295	1.3234	1.5587	46.9481
Low, Volume, 0-C	0.9758	0.8276	0.9878	29.5070
Low, Pre, Close, 0-C	1.2222	0.8340	1.1056	29.6586
Low, H-L, 0-C	3.3821	1.4324	1.8390	50.8115
Low, 0-C, SMA_7	2.5383	1.5413	1.5932	54.5860
Low, 0-C, rsi_7	5.7383	2.2097	2.3955	78.5479
Low, 0-C, EMA_7	0.5525	0.5667	0.7433	19.8413
Open, High, EMA_7	1.0161	0.8580	1.0080	30.0663
Open, Low, EMA_7	9.7047	2.9833	3.1152	105.3294
Open, Volume, EMA_7	0.5865	0.7030	0.7658	24.9025
Open, Pre, Close, EMA_7	1.1259	1.1259	1.0611	31.9841
Open, H-L, EMA_7	1.5746	1.0215	1.2548	36.5010
Open, 0-C, EMA_7	3.7429	1.8581	1.9347	65.9506
Open, SMA_7, EMA_7	**0.4430**	**0.5100**	**0.6656**	**17.8809**
Open, rsi_7, EMA_7	2.1048	1.3308	1.4508	46.8902
Open, High, SMA_7	1.9173	1.3435	1.3847	47.6489
High, Low, SMA_7	**0.4116**	**0.5410**	**0.6415**	**19.2410**
High, Volume, SMA_7	1.3964	0.8987	1.1817	32.1054
High, Pre, Close, SMA_7	1.7762	1.2380	1.3327	43.7537
High, H-L, SMA_7	2.4195	1.4988	1.5555	53.1760
High, 0-C, SMA_7	4.7892	2.1033	2.1884	74.4599
High, SMA_7, rsi_7	1.6358	1.1959	1.2790	42.0827
High, SMA_7, EMA_7	**0.4942**	**0.5391**	**0.7030**	**19.1890**
Open, High, 0-C	3.2579	1.6257	1.8050	57.4481
High, Low, 0-C	2.8196	1.5275	1.6792	54.1375
High, Volume, 0-C	1.2652	0.8335	1.1248	29.7262
High, Pre, Close, 0-C	0.8101	0.7805	0.9001	27.3142
High, H-L, 0-C	13.6154	3.4139	3.6899	120.7008
High, 0-C, SMA_7	5.0254	2.0855	2.2417	73.6498

High, 0-C, rsi_7	2.1319	1.4340	1.4601	50.7138
High, 0-C, EMA_7	3.5155	1.7405	1.8750	61.8524

Note: Bolded values indicate the 5 best results(3 features).The 5 best results features will be selected for the next CNN experiments by adding up 1 feature.

Table 22 above summarizes the results of SAND Prediction by CNN model in Experiment 6. In experiment 6 utilising CNN as the predictive model, "High, rsi 7, EMA_7", "SMA_7, rsi_7, EMA_7", "Open, SMA_7, EMA_7", "Volume, 0-C, EMA 7", and "Volume, Pre_Close, rsi_7" are the top five three-feature combinations with the lowest MAPE values, which are 11.8246, 19.0058 And among these characteristics, "High, rsi 7, and EMA 7" have the greatest predictive power, as the average absolute percentage error of forecasts is 11.8246 % lowest.

6.1.7 Experiment 7 (4 features)

The experiment 7 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in this section implicitly. Experiment 7 uses four dependent features as input to predict the next ten days of closing price, labelled as “Close” target features. Experiment 7 evaluates the predictability of four-feature combinations by adding one more feature individually to the three selective four-feature combinations with the lowest MAPE value in experiment 6. In experiment 7, there are 32 test cases for LSTM, GRUs models and 33 test cases for CNN. SAND will be the primary metaverse cryptocurrencies used to evaluate the predictivity of the four-feature combinations. Five selective four-feature combinations with the lowest MSE, MAE, MSE and MAPE values will be selected for the following experiments.

6.1.7.1 LSTM

Table 23: Results of SAND Prediction by LSTM model in Experiment 7

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Pre_Close, EMA_7	**5.7387**	**1.9190**	**2.3956**	**67.9587**
High, Low, Pre_Close, EMA_7	**4.4529**	**1.7931**	**2.1102**	**63.3758**
High, Volume, Pre_Close, EMA_7	9.5060	2.9890	3.0832	106.1332
High, Pre_Close, H-L, EMA_7	18.5466	4.2978	4.3066	152.1118
High, Pre_Close, 0-C, EMA_7	8.0956	2.7368	2.8453	97.1580
High, Pre_Close, SMA_7, EMA_7	8.2779	2.4759	2.8771	87.2566
High, Pre_Close, rsi_7, EMA_7	9.7213	2.3450	3.1179	83.7297
Open, High, Low, Pre_Close	5.7406	2.2133	2.3959	78.5632

High, Volume, Low, Pre_Close	9.1005	2.9237	3.0167	103.7745
High, Low, Pre_Close, H-L	**3.0101**	**1.5030**	**1.7350**	**53.5293**
High, Low, Pre_Close, 0-C	4.8235	2.0564	2.1962	73.0447
High, Low, Pre_Close, SMA_7	7.4343	2.3171	2.7266	81.7557
High, Low, Pre_Close, rsi_7	8.1696	2.6257	2.8583	93.3326
Open, High, Low, SMA_7	7.4975	2.3382	2.7382	82.5067
Open, High, Volume, SMA_7	10.6857	3.2521	3.2689	115.1050
Open, High, Pre_Close, SMA_7	8.0516	2.3837	2.8375	84.1905
Open, High, H-L, SMA_7	7.1703	2.6566	2.6777	94.1773
Open, High, 0-C, SMA_7	7.5427	2.5435	2.7464	90.3982
Open, High, SMA_7, rsi_7	13.1852	3.3588	3.6311	119.3900
Open, High, SMA_7, EMA_7	10.5654	2.4940	3.2505	88.5548
Open, Low, Volume, Pre_Close	15.1675	3.8860	3.8945	137.5462
Open, Low, Pre_Close, H-L	7.3777	2.7056	2.7162	95.7635
Open, Low, Pre_Close, 0-C	5.1226	2.1398	2.2633	75.9486
Open, Low, Pre_Close, SMA_7	**1.6792**	**1.0794**	**1.2959**	**38.4433**
Open, Low, Pre_Close, rsi_7	7.8715	2.6854	2.8056	95.3320
Open, Low, Pre_Close, EMA_7	**4.3410**	**1.6503**	**2.0835**	**58.6924**
High, Low, Volume, SMA_7	11.3544	3.3654	3.3696	119.1061
High, Low, Pre_Close, SMA_7	8.0146	2.4348	2.8310	85.8585
High, Low, SMA_7, H-L	12.5258	3.5308	3.5392	124.9374
High, Low, SMA_7, 0-C	6.7702	2.5450	2.6020	90.2304
High, Low, SMA_7, rsi_7	8.9701	2.7254	2.9950	96.8566
High, Low, SMA_7, EMA_7	11.0182	2.8352	3.3194	100.1808

Note: Bolded values indicate the 5 best results(4 features).The 5 best results features will be selected for the next LSTM experiments by adding up 1 feature.

Table 23 above summarizes the results of SAND Prediction by LSTM model in Experiment 7. In experiment 7 using LSTM as the predictive model, "Open, High, Pre_Close, EMA_7", "High, Low, Pre_Close, EMA_7", "High, Low, Pre_Close, H-L", "Open, Low, Pre_Close, SMA_7" and "Open, High, Pre_Close, EMA_7" are the top 5 four-feature combinations that have the top 5 lowest MAPE values which are recorded as 67.9587, 63.3758, 53.5293, 38.4433 and 58.6924. Furthermore, among these features, the "Open, Low, Pre_Close, SMA_7" have the highest predictive power; the lowest average of the absolute percentage errors of forecasts is recorded as 38.4433%.

6.1.7.2 GRUs

Table 24: Results of SAND Prediction by GRUs model in Experiment 7

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Volume, H-L	0.7510	0.6848	0.8666	24.0901
Open, Low, Volume, H-L	3.6063	1.8207	1.8990	64.6550
Open, Volume, Pre_Close, H-L	2.3807	1.4155	1.5430	50.3482
Open, Volume, H-L, 0-C	3.0440	1.6478	1.7447	58.4978
Open, Volume, H-L, SMA_7	2.3009	1.2837	1.5169	45.7762

Open, Volume, H-L, rsi_7	**0.2995**	**0.4337**	**0.5473**	**15.2728**
Open, Volume, H-L, EMA_7	3.4266	1.6865	1.8511	59.8625
Open, High, H-L, rsi_7	2.3204	1.4066	1.5233	49.5256
Open, Low, H-L, rsi_7	**0.3690**	**0.5514**	**0.6075**	**19.3907**
Open, Pre_Close, H-L, rsi_7	0.4373	0.6113	0.6613	21.4820
Open, H-L, 0-C, rsi_7	0.5428	0.5471	0.7368	19.5736
Open, H-L, SMA_7, rsi_7	0.7036	0.6885	0.8388	24.1286
Open, H-L, rsi_7, EMA_7	2.5081	1.5032	1.5837	52.9144
Open, High, Pre_Close, H-L	0.7184	0.7713	0.8476	27.1020
Open, Low, Pre_Close, H-L	1.3780	1.0828	1.1739	38.4893
Open, Pre_Close, H-L, 0-C	1.4239	1.0674	1.1933	37.8871
Open, Pre_Close, H-L, SMA_7	**0.4126**	**0.5177**	**0.6423**	**18.2036**
Open, Pre_Close, H-L, EMA_7	0.5478	0.6402	0.7402	22.4904
Open, Volume, 0-C, EMA_7	0.5316	0.5774	0.7291	20.3452
High, Volume, 0-C, EMA_7	5.0996	2.2117	2.2582	77.9430
Low, Volume, 0-C, EMA_7	0.5416	0.6517	0.7360	22.9144
Volume, Pre_Close, 0-C, EMA_7	**0.3249**	**0.4688**	**0.5700**	**16.5677**
Volume, H-L, 0-C, EMA_7	2.4147	1.5192	1.5539	53.6190
Volume, 0-C, SMA_7, EMA_7	1.7881	1.0713	1.3372	37.8293
Volume, 0-C, rsi_7, EMA_7	0.6067	0.6913	0.7789	24.2407
Open, Volume, Pre_Close, rsi_7	6.1881	2.3228	2.4876	82.6536
High, Volume, Pre_Close, rsi_7	**0.4561**	**0.5792**	**0.6753**	**20.2362**
Low, Volume, Pre_Close, rsi_7	5.4837	2.2016	2.3417	78.3238
Volume, Pre_Close, H-L, rsi_7	3.0484	1.5987	1.7460	56.9608
Volume, Pre_Close, 0-C, rsi_7	7.3822	2.6095	2.7170	92.7441
Volume, Pre_Close, SMA_7, rsi_7	1.7603	1.1580	1.3268	41.2139
Volume, Pre_Close, rsi_7, EMA_7	1.7212	1.2378	1.3119	43.4803

Note: Bolded values indicate the 5 best results(4 features).The 5 best results features will be selected for the next GRUs experiments by adding up 1 feature.

Table 24 above summarizes the results of SAND Prediction by GRUs model in Experiment 7. In experiment 7 using GRUs as the predictive model, "Open, Volume, H-L, rsi_7", "Open, Low, H-L, rsi_7", "Open, Pre_Close, H-L, SMA_7", "Volume, Pre_Close, 0-C, EMA_7" and "High, Volume, Pre_Close, rsi_7" are the top 5 four-feature combinations that have the top 5 lowest MAPE values which are recorded as 15.2728, 19.3907, 18.2036, 16.5677 and 20.2362. And among these features, the "Volume, Pre_Close, 0-C, EMA_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 16.5677%.

6.1.7.3 CNN

Table 25: Results of SAND Prediction by CNN model in Experiment 7

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, rsi_7, EMA_7	0.6244	0.7118	0.7902	25.1923
High, Low, rsi_7, EMA_7	2.5918	1.5077	1.6099	53.5097
High, Volume, rsi_7, EMA_7	**0.5345**	**0.6305**	**0.7311**	**22.1473**

High, Pre, Close, rsi_7, EMA_7	**0.5340**	**0.6335**	**0.7307**	**22.3761**
High, H-L, rsi_7, EMA_7	1.9652	1.1642	1.4019	41.0592
High, 0-C, rsi_7, EMA_7	1.1723	0.9374	1.0827	33.3085
High, SMA_7, rsi_7, EMA_7	0.6714	0.7348	0.8194	25.9705
Open, High, SMA_7, EMA_7	**0.2654**	**0.4295**	**0.5151**	**15.1814**
Open, Low, SMA_7, EMA_7	4.2388	1.9078	2.0588	67.0221
Open, Volume, SMA_7, EMA_7	**0.2280**	**0.4088**	**0.4774**	**14.4494**
Open, Pre, Close, SMA_7, EMA_7	3.0381	1.6642	1.7430	58.8716
Open, H-L, SMA_7, EMA_7	1.1577	0.9200	1.0760	32.5037
Open, 0-C, SMA_7, EMA_7	0.8097	0.8280	0.8998	29.3236
Open, SMA_7, rsi_7, EMA_7	0.5118	0.6653	0.7154	23.3850
High, SMA_7, rsi_7, EMA_7	**0.5143**	**0.6180**	**0.7172**	**21.8731**
Low, SMA_7, rsi_7, EMA_7	1.1281	0.9046	1.0621	31.8932
Volume, SMA_7, rsi_7, EMA_7	4.2764	1.6185	2.0680	57.1896
Pre, Close, SMA_7, rsi_7, EMA_7	0.8404	0.8758	0.9167	31.0742
H-L, SMA_7, rsi_7, EMA_7	0.9055	0.8445	0.9516	29.8411
0-C, SMA_7, rsi_7, EMA_7	4.2839	1.9946	2.0698	70.6228
High, Low, SMA_7, EMA_7	2.2316	1.3512	1.4939	47.9842
High, Volume, SMA_7, EMA_7	0.9773	0.9001	0.9886	31.7931
High, Pre, Close, SMA_7, EMA_7	1.0459	0.8584	1.0227	30.3676
High, H-L, SMA_7, EMA_7	0.7624	0.6426	0.8731	22.9707
High, 0-C, SMA_7, EMA_7	2.1608	1.2675	1.4700	45.1025
Open, High, Low, SMA_7	1.5329	0.8247	1.2381	29.2990
High, Low, Volume, SMA_7	4.2572	2.0079	2.0633	71.0368
High, Low, Pre, Close, SMA_7	10.6800	3.0264	3.2680	107.7288
High, Low, H-L, SMA_7	0.6656	0.7294	0.8159	25.5390
High, Low, 0-C, SMA_7	0.5629	0.6319	0.7503	22.3969
High, Low, SMA_7, rsi_7	0.6381	0.7084	0.7988	25.2325
Open, High, rsi_7, EMA_7	0.6244	0.7118	0.7902	25.1923
High, Low, rsi_7, EMA_7	2.5918	1.5077	1.6099	53.5097

Note: Bolded values indicate the 5 best results(4 features).The 5 best results features will be selected for the next CNN experiments by adding up 1 feature

Table 25 above summarizes the results of SAND Prediction by CNN model in Experiment 7. In experiment 7 using CNN as the predictive model, "High, Volume, rsi_7, EMA_7", "High, Pre, Close, rsi_7, EMA_7", "Open, High, SMA_7, EMA_7", "Open, Volume, SMA_7, EMA_7" and "High, SMA_7, rsi_7, EMA_7" are the top 5 four-feature combinations that have the top 5 lowest MAPE value which is recorded as 22.1473, 22.3761, 15.1814, 14.4494 and 21.8731. Furthermore, among these features, the "Open, Volume, SMA_7, EMA_7" have the highest predictive power; the lowest average of the absolute percentage errors of forecasts is recorded as 14.4494%.

6.1.8 Experiment 8 (5 features)

The experiment 8 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in this section implicitly. Experiment 8 uses five dependent features as input to predict the next ten days of closing price, labelled as "Close" target features. Experiment 8 evaluates the predictability of five-feature combinations by adding one more feature individually to the five selective four-feature combinations with the lowest MAPE value in experiment 7. For experiment 8, there are 26,29 and 25 test cases for each model: LSTM, GRUs and CNN. SAND will be the primary metaverse cryptocurrencies used to evaluate the predictivity of the five-feature combinations. Five selective five-feature combinations with the lowest MSE, MAE, MSE and MAPE values will be selected for the following experiments.

6.1.8.1 LSTM

Table 26: Results of SAND Prediction by LSTM model in Experiment 8

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Low, Pre_Close, SMA_7	**7.0036**	**2.0874**	**2.6464**	**74.1248**
Open, Low, Volume, Pre_Close, SMA_7	12.5605	3.5370	3.5441	125.1462
Open, Low, Pre_Close, H-L, SMA_7	12.9864	3.5864	3.6037	126.8137
Open, Low, Pre_Close, 0-C, SMA_7	7.2707	2.6731	2.6964	94.6871
Open, Low, Pre_Close, rsi_7, SMA_7	9.2624	2.7990	3.0434	99.4734
Open, Low, Pre_Close, SMA_7, EMA_7	**6.0727**	**2.0618**	**2.4643**	**73.4913**
Open, High, Low, Pre_Close, H-L	8.3982	2.8533	2.8980	101.1886
High, Low, Volume, Pre_Close, H-L	11.9043	3.4448	3.4503	121.9513
High, Low, Pre_Close, H-L, 0-C	8.9703	2.9908	2.9950	105.8446
High, Low, Pre_Close, H-L, SMA_7	11.4900	3.3756	3.3897	119.5775
High, Low, Pre_Close, H-L, rsi_7	7.9108	2.7701	2.8126	98.1838
Open, High, Low, Pre_Close, EMA_7	**4.6650**	**1.8294**	**2.1599**	**64.5149**
Open, Low, Volume, Pre_Close, EMA_7	12.6935	3.5555	3.5628	125.8617
Open, Low, Pre_Close, H-L, EMA_7	12.2144	3.4846	3.4949	123.4025
Open, Low, Pre_Close, 0-C, EMA_7	6.5905	2.5320	2.5672	89.7341
Open, Low, Pre_Close, rsi_7, EMA_7	10.9666	3.0467	3.3116	108.3118
High, Low, Volume, Pre_Close, EMA_7	**5.0246**	**1.8852**	**2.2416**	**67.2055**
High, Low, Pre_Close, H-L, EMA_7	10.5667	3.1096	3.2507	110.4861
High, Low, Pre_Close, 0-C, EMA_7	5.9600	2.3923	2.4413	84.8219
High, Low, Pre_Close, SMA_7, EMA_7	7.9822	2.3773	2.8253	83.9269
High, Low, Pre_Close, rsi_7, EMA_7	11.3754	3.0757	3.3727	109.3639
Open, High, Volume, Pre_Close, EMA_7	9.5839	3.0610	3.0958	108.5270
Open, High, Pre_Close, H-L, EMA_7	6.4137	2.3204	2.5325	82.6054
Open, High, Pre_Close, 0-C, EMA_7	6.9566	2.5041	2.6375	88.9450
Open, High, Pre_Close, SMA_7, EMA_7	**6.9381**	**2.1223**	**2.6340**	**75.1058**
Open, High, Pre_Close, rsi_7, EMA_7	10.4091	2.8269	3.2263	100.6494

Note: Bolded values indicate the 5 best results(5 features).The 5 best results features will be selected for the next LSTM experiments by adding up 1 feature.

Table 26 above summarizes the results of SAND Prediction by LSTM model in Experiment 8. In experiment 8 using LSTM as the predictive model, Open, High, Low, Pre_Close, EMA_7", "Open, Low, Pre_Close, SMA_7, EMA_7", "Open, High, Low, Pre_Close, EMA_7", "High, Low, Volume, Pre_Close, EMA_7" and "Open, High, Pre_Close, SMA_7, EMA_7" are the top 5 five-feature combinations that have the top 5 lowest MAPE values which are recorded as 74.1248, 73.4913, 64.5149, 67.2055 and 75.1058. Furthermore, among these features, the "Open, High, Low, Pre_Close, EMA_7" have the highest predictive power; the lowest average of the absolute percentage errors of forecasts is recorded as 64.5149 %.

6.1.8.2 GRUs

Table 27: Results of SAND Prediction by GRUs model in Experiment 8

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Volume, H-L, rsi_7	0.4016	0.5498	0.6337	19.2649
Open, Low, Volume, H-L, rsi_7	1.1952	0.9809	1.0932	34.9615
Open, Volume, Pre_Close, H-L, rsi_7	**0.3031**	**0.4290**	**0.5505**	**15.1609**
Open, Volume, H-L, 0-C, rsi_7	**0.3269**	**0.4340**	**0.5717**	**15.4700**
Open, Volume, H-L, SMA_7, rsi_7	0.7013	0.7107	0.8375	24.9285
Open, Volume, H-L, rsi_7, EMA_7	0.7831	0.7999	0.8849	28.0714
Open, High, H-L, 0-C, rsi_7	2.2713	1.4698	1.5071	51.8255
Open, Low, H-L, 0-C, rsi_7	0.9399	0.7815	0.9695	27.9356
Open, Pre_Close, H-L, 0-C, rsi_7	**0.5456**	**0.5252**	**0.7387**	**18.6988**
Open, H-L, 0-C, SMA_7, rsi_7	1.1057	1.0231	1.0515	36.1321
Open, H-L, 0-C, rsi_7, EMA_7	3.2074	1.7552	1.7909	61.9431
Open, High, Pre_Close, H-L, SMA_7	**0.4338**	**0.5304**	**0.6586**	**18.6181**
Open, Low, Pre_Close, H-L, SMA_7	2.1918	1.3459	1.4805	47.9303
Open, Volume, Pre_Close, H-L, SMA_7	0.5825	0.6435	0.7632	22.6314
Open, Pre_Close, H-L, 0-C, SMA_7	0.4963	0.5664	0.7045	19.8949
Open, Pre_Close, H-L, SMA_7, rsi_7	0.6193	0.6806	0.7869	23.8790
Open, Pre_Close, H-L, SMA_7, EMA_7	1.5040	1.0987	1.2264	38.6139
Open, Volume, Pre_Close, 0-C, EMA_7	0.4879	0.5620	0.6985	20.0056
High, Volume, Pre_Close, 0-C, EMA_7	1.1122	0.9635	1.0546	33.8711
Low, Volume, Pre_Close, 0-C, EMA_7	0.9097	0.9097	0.9538	27.2372
Volume, Pre_Close, H-L, 0-C, EMA_7	3.0112	1.6600	1.7353	58.9522
Volume, Pre_Close, 0-C, SMA_7, EMA_7	1.1059	0.8895	1.0516	31.1371
Volume, Pre_Close, 0-C, rsi_7, EMA_7	1.5976	1.0398	1.2640	36.9716
Open, High, Volume, Pre_Close, rsi_7	1.2302	1.0197	1.1091	35.7694
High, Low, Volume, Pre_Close, rsi_7	0.5511	0.6085	0.7423	21.2353
High, Volume, Pre_Close, H-L, rsi_7	0.5548	0.6521	0.7449	22.8067

High, Volume, Pre_Close, 0-C, rsi_7	**0.2570**	**0.4131**	**0.5070**	**14.4718**
High, Volume, Pre_Close, SMA_7, rsi_7	1.0420	0.8445	1.0208	29.5086
High, Volume, Pre_Close, rsi_7, EMA_7	0.6925	0.7226	0.8322	25.2748

Note: Bolded values indicate the 5 best results(5 features).The 5 best results features will be selected for the next GRUs experiments by adding up 1 feature.

Table 27 above summarizes the results of SAND Prediction by GRUs model in Experiment 8. In experiment 8 using GRUs as the predictive model, "Open, Volume, Pre_Close, H-L, rsi_7", "Open, Volume, H-L, 0-C, rsi_7", "Open, Pre_Close, H-L, 0-C, rsi_7", "Open, High, Pre_Close, H-L, SMA_7" and "High, Volume, Pre_Close, 0-C, rsi_7" are the top 5 five-feature combinations that have the top 5 lowest MAPE values which are recorded as 15.1609, 15.47, 18.6988, 18.6181 and 14.4718. Moreover, among these features, the High, "Volume, Pre_Close, 0-C, rsi_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 14.4718 %.

6.1.8.3 CNN

Table 28: Results of SAND Prediction by CNN model in Experiment 8

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Volume, SMA_7, EMA_7	1.0797	0.8304	1.0391	29.3310
Open, Low, Volume, SMA_7, EMA_7	1.3926	0.8790	1.1801	31.1881
Open, Volume, Pre_Close, SMA_7, EMA_7	0.6814	0.6544	0.8255	23.1661
Open, Volume, H-L, SMA_7, EMA_7	3.1080	1.5059	1.7629	53.0992
Open, Volume, 0-C, SMA_7, EMA_7	0.5853	0.5723	0.7650	20.3284
Open, Volume, SMA_7, rsi_7, EMA_7	**0.5450**	**0.5462**	**0.7383**	**19.4010**
Open, High, Low, SMA_7, EMA_7	1.3818	0.9913	1.1755	35.3046
Open, High, Pre_Close, SMA_7, EMA_7	0.9742	0.7113	0.9870	24.9440
Open, High, H-L, SMA_7, EMA_7	1.7725	1.1606	1.3314	40.8941
Open, High, 0-C, SMA_7, EMA_7	**0.3744**	**0.4419**	**0.6119**	**15.6250**
Open, High, SMA_7, rsi_7, EMA_7	**0.2952**	**0.4614**	**0.5433**	**16.4575**
High, Low, SMA_7, rsi_7, EMA_7	0.6482	0.6500	0.8051	23.2619
High, Volume, SMA_7, rsi_7, EMA_7	0.3974	0.5827	0.6304	20.5563
High, Pre_Close, SMA_7, rsi_7, EMA_7	**0.1788**	**0.3912**	**0.4228**	**13.8874**
High, H-L, SMA_7, rsi_7, EMA_7	0.7749	0.8095	0.8803	28.7319
High, 0-C, SMA_7, rsi_7, EMA_7	1.5815	0.9150	1.2576	33.0663
Open, High, Volume, rsi_7, EMA_7	1.5815	0.9150	1.2576	32.4591
High, Low, Volume, rsi_7, EMA_7	1.8079	0.8891	1.3446	31.5683
High, Volume, Pre_Close, rsi_7, EMA_7	0.9205	0.8038	0.9594	28.1661
High, Volume, H-L, rsi_7, EMA_7	2.4723	1.3960	1.5724	49.2090
High, Volume, 0-C, rsi_7, EMA_7	1.0541	0.8047	1.0267	28.5408
Open, High, Pre_Close, rsi_7, EMA_7	1.0891	0.9326	1.0436	33.1650
High, Low, Pre_Close, rsi_7, EMA_7	1.0720	0.9319	1.0354	33.0044
High, Pre_Close, H-L, rsi_7, EMA_7	0.8922	0.7721	0.9446	27.3642
High, Pre_Close, 0-C, rsi_7, EMA_7	**0.1550**	**0.3304**	**0.3937**	**11.6837**

Note: Bolded values indicate the 5 best results(5 features).The 5 best results features will be selected for the next CNN experiments by adding up 1 feature.

Table 28 above summarizes the results of SAND Prediction by CNN model in Experiment 8. In experiment 8 using GRUs as the predictive model, "Open, Volume, SMA_7, rsi_7, EMA_7", "Open, High, 0-C, SMA_7, EMA_7", "Open, High, SMA_7, rsi_7, EMA_7", "High, Pre_Close, SMA_7, rsi_7, EMA_7" and "High, Pre_Close, 0-C, rsi_7, EMA_7" are the top 5 five-feature combinations that have the top 5 lowest MAPE values which are recorded as 19.401, 19.401, 16.4575, 13.8874 and 11.6837. Moreover, among these features, the "High, Pre_Close, 0-C, rsi_7, EMA_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 11.6837 %.

6.1.9 Experiment 9 (6 features)

The experiment 9 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in this section implicitly. Experiment 9 uses six dependent features as input to predict the next ten days of closing price, labelled as "Close" target features. Experiment 9 evaluates the predictability of six-feature combinations by adding one more feature individually to the five selective five-feature combinations with the lowest MAPE value in experiment 8. In experiment 9, there are 22 test cases for LSTM, CNN models and 23 test cases for GRUs. SAND will be the primary metaverse cryptocurrencies used to evaluate the predictivity of the six-feature combinations. Five selective six-feature combinations with the lowest MSE, MAE, MSE and MAPE values will be selected for the following experiments.

6.1.9.1 LSTM

Table 29: Results of SAND Prediction by LSTM model in Experiment 9

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Low, Volume, Pre_Close, EMA_7	7.6962	2.6873	2.7742	95.3672
Open, High, Low, Pre_Close, H-L, EMA_7	9.6051	3.0316	3.0992	107.5416
Open, High, Low, Pre_Close, 0-C, EMA_7	6.6072	2.4604	2.5705	87.3453
Open, High, Low, Pre_Close, SMA_7, EMA_7	**6.7147**	**2.0682**	**2.5913**	**73.3435**
Open, High, Low, Pre_Close, rsi_7, EMA_7	**8.5317**	**2.3082**	**2.9209**	**82.3686**
Open, High, Low, Volume, Pre_Close, EMA_7	7.6962	2.6873	2.7742	95.3672

High, Low, Volume, Pre_Close, H-L, EMA_7	13.1109	3.5996	3.6209	127.5203
High, Low, Volume, Pre_Close, 0-C, EMA_7	10.1658	3.1365	3.1884	111.1888
High, Low, Volume, Pre_Close, SMA_7, EMA_7	**6.1734**	**2.1835**	**2.4846**	**77.8195**
High, Low, Volume, Pre_Close, rsi_7, EMA_7	11.3387	3.3052	3.3673	117.2082
Open, Low, Volume, Pre_Close, SMA_7, EMA_7	10.0326	3.0684	3.1674	108.9129
Open, Low, Pre_Close, H-L, SMA_7, EMA_7	**4.3287**	**1.9669**	**2.0806**	**69.8861**
Open, Low, Pre_Close, 0-C, SMA_7, EMA_7	7.9977	2.8035	2.8280	99.2679
Open, Low, Pre_Close, SMA_7, rsi_7, EMA_7	12.0452	2.9298	3.4706	104.4487
Open, High, Low, Volume, Pre_Close, SMA_7	6.9819	2.4089	2.6423	85.7404
Open, High, Low, Pre_Close, H-L, SMA_7	11.5999	3.3691	3.4059	119.3999
Open, High, Low, Pre_Close, 0-C, SMA_7	6.5767	2.4672	2.5645	87.5356
Open, High, Low, Pre_Close, SMA_7, rsi_7	13.5172	3.4450	3.6766	122.3720
Open, High, Volume, Pre_Close, SMA_7, EMA_7	**6.3755**	**2.2858**	**2.5250**	**81.4098**
Open, High, Pre_Close, H-L, SMA_7, EMA_7	7.7725	2.7133	2.7879	96.3052
Open, High, Pre_Close, 0-C, SMA_7, EMA_7	8.0444	2.7684	2.8363	98.0969
Open, High, Pre_Close, SMA_7, rsi_7, EMA_7	9.7373	2.4708	3.1205	87.9172

Note: Bolded values indicate the 5 best results(6 features).The 5 best results features will be selected for the next LSTM experiments by adding up 1 feature.

Table 29 above summarizes the results of SAND Prediction by LSTM model in Experiment 9. In experiment 9 using LSTM as the predictive model, "Open, High, Low, Pre_Close, SMA_7, EMA_7", "Open, High, Low, Pre_Close, rsi_7, EMA_7", "High, Low, Volume, Pre_Close, SMA_7, EMA_7", "Open, Low, Pre_Close, H-L, SMA_7, EMA_7" and "Open, High, Volume, Pre_Close, SMA_7, EMA_7" are the top 5 six-feature combinations that have the top 5 lowest MAPE values which are recorded as 73.3435, 82.3686, 77.8195, 69.8861 and 81.4098. Moreover, among these features, the "Open, Low, Pre_Close, H-L, SMA_7, EMA_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 69.8861%.

6.1.9.2 GRUs

Table 30: Results of SAND Prediction by GRUs model in Experiment 9

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Volume, Pre_Close, H-L, rsi_7	1.0868	0.8187	1.0425	29.1846
Open, Low, Volume, Pre_Close, H-L, rsi_7	0.6968	0.7249	0.8347	25.3739
Open, Volume, Pre_Close, H-L, 0-C, rsi_7	**0.1689**	**0.3028**	**0.4110**	**10.7296**
Open, Volume, Pre_Close, H-L, SMA_7, rsi_7	0.8174	0.8236	0.9041	28.8942
Open, Volume, Pre_Close, H-L, rsi_7, EMA_7	**0.3316**	**0.4527**	**0.5758**	**15.7601**
Open, High, Volume, H-L, 0-C, rsi_7	2.1753	1.4664	1.4749	51.7232
Open, Low, Volume, H-L, 0-C, rsi_7	**0.0795**	**0.2443**	**0.2820**	**8.5567**
Open, Volume, H-L, 0-C, SMA_7, rsi_7	0.6155	0.7032	0.7846	24.7687

Open, Volume, H-L, 0-C, rsi_7, EMA_7	0.9627	0.8908	0.9812	31.3060
Open, High, Pre_Close, H-L, 0-C, rsi_7	1.0351	0.9795	1.0174	34.4955
Open, Low, Pre_Close, H-L, 0-C, rsi_7	2.3908	1.3558	1.5462	48.3723
Open, Pre_Close, H-L, 0-C, SMA_7, rsi_7	0.9677	0.8201	0.9837	29.0900
Open, Pre_Close, H-L, 0-C, rsi_7, EMA_7	0.4852	0.5792	0.6966	20.3079
Open, High, Low, Pre_Close, H-L, SMA_7	0.4656	0.5653	0.6823	19.8719
Open, High, Volume, Pre_Close, H-L, SMA_7	1.2214	0.8820	1.1051	31.4800
Open, High, Pre_Close, H-L, 0-C, SMA_7	0.4931	0.5379	0.7022	18.9661
Open, High, Pre_Close, H-L, SMA_7, rsi_7	1.7411	1.1546	1.3195	40.4799
Open, High, Pre_Close, H-L, SMA_7, EMA_7	1.3071	1.0350	1.1433	36.4080
Open, High, Volume, Pre_Close, 0-C, rsi_7	**0.2718**	**0.4187**	**0.5214**	**14.5845**
Low, High, Volume, Pre_Close, 0-C, rsi_7	**0.1667**	**0.3272**	**0.4083**	**11.3706**
High, Volume, Pre_Close, H-L, 0-C, rsi_7	0.5339	0.6959	0.7307	24.4585
High, Volume, Pre_Close, 0-C, SMA_7, rsi_7	0.3352	0.4831	0.5790	16.9656
High, Volume, Pre_Close, 0-C, rsi_7, EMA_7	1.0943	1.0255	1.0461	36.1017

Note: Bolded values indicate the 5 best results(6 features).The 5 best results features will be selected for the next GRUs experiments by adding up 1 feature.

Table 30 above summarizes the results of SAND Prediction by GRUs model in Experiment 9. In experiment 9 using GRUs as the predictive model, "Open, Volume, Pre_Close, H-L, 0-C, rsi_7", "Open, Volume, Pre_Close, H-L, rsi_7, EMA_7", "Open, Low, Volume, H-L, 0-C, rsi_7", "Open, High, Volume, Pre_Close, 0-C, rsi_7" and "Low, High, Volume, Pre_Close, 0-C, rsi_7" are the top 5 six-feature combinations that have the top 5 lowest MAPE values which are recorded as 10.7296, 15.7601, 8.5567, 14.5845 and 11.3706. Moreover, among these features, the "Open, Low, Volume, H-L, 0-C, rsi_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 8.5567%.

6.1.9.3 CNN

Table 31: Results of SAND Prediction by CNN model in Experiment 9

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Pre_Close, 0-C, rsi_7, EMA_7	**0.2970**	**0.4074**	**0.5449**	**14.4314**
High, Low, Pre_Close, 0-C, rsi_7, EMA_7	**0.2408**	**0.3888**	**0.4907**	**13.6724**
High, Volume, Pre_Close, 0-C, rsi_7, EMA_7	1.5499	1.0579	1.2450	37.0702
High, Pre_Close, H-L, 0-C, rsi_7, EMA_7	3.1552	1.6216	1.7763	57.4519
High, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7	1.3875	1.1103	1.1779	39.1750
Open, High, Pre_Close, SMA_7, rsi_7, EMA_7	1.9406	1.2551	1.3930	44.4612
High, Low, Pre_Close, SMA_7, rsi_7, EMA_7	4.2569	1.8309	2.0632	65.1884
High, Volume, Pre_Close, SMA_7, rsi_7, EMA_7	**0.6728**	**0.7038**	**0.8203**	**25.1104**
High, Pre_Close, H-L, SMA_7, rsi_7, EMA_7	2.8901	1.4260	1.7000	50.8086
Open, High, Volume, SMA_7, rsi_7, EMA_7	2.2529	1.3870	1.5010	49.3251
Open, Low, Volume, SMA_7, rsi_7, EMA_7	3.8165	1.8569	1.9536	65.5989
Open, Volume, Pre_Close, SMA_7, rsi_7, EMA_7	0.7080	0.7584	0.8415	26.6975
Open, Volume, H-L, SMA_7, rsi_7, EMA_7	2.3553	1.2795	1.5347	45.4530

Open, Volume, 0-C, SMA_7, rsi_7, EMA_7	0.8613	0.8017	0.9281	28.4376
Open, High, Low, 0-C, SMA_7, EMA_7	0.8380	0.7802	0.9154	27.5059
Open, High, Volume, 0-C, SMA_7, EMA_7	4.8182	2.0018	2.1950	71.1950
Open, High, Pre_Close, 0-C, SMA_7, EMA_7	**0.2289**	**0.3733**	**0.4785**	**13.1264**
Open, High, H-L, 0-C, SMA_7, EMA_7	5.6317	2.2157	2.3731	79.0039
Open, High, 0-C, SMA_7, rsi_7, EMA_7	0.8791	0.8624	0.9376	30.3429
Open, High, Low, SMA_7, rsi_7, EMA_7	0.6806	0.7152	0.8250	25.3118
Open, High, H-L, SMA_7, rsi_7, EMA_7	1.1222	0.9178	1.0593	32.0926
Open, High, 0-C, SMA_7, rsi_7, EMA_7	**0.3779**	**0.5256**	**0.6147**	**18.6078**

Note: Bolded values indicate the 5 best results(6 features).The 5 best results features will be selected for the next CNN experiments by adding up 1 feature.

Table 31 above summarizes the results of SAND Prediction by CNN model in Experiment 9. In experiment 9 using CNN as the predictive model, "Open, High, Pre_Close, 0-C, rsi_7, EMA_7", "High, Low, Pre_Close, 0-C, rsi_7, EMA_7","High, Volume, Pre_Close, SMA_7, rsi_7, EMA_7", "Open, High, Pre_Close, 0-C, SMA_7, EMA_7" and "Open, High, 0-C, SMA_7, rsi_7, EMA_7" are the top 5 six-feature combinations that have the top 5 lowest MAPE values which are recorded as 14.4314, 13.6724, 25.1104, 13.1264 and 18.6078. Moreover, among these features, the "Open, High, Pre_Close, 0-C, SMA_7, EMA_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 13.1264%.

6.1.10 Experiment 10 (7 features)

The experiment 10 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in this section implicitly. Experiment 10 uses seven dependent features as input to predict the next ten days of closing price, labelled as "Close" target features. Experiment 10 evaluates the predictability of seven-feature combinations by adding one more feature individually to the five selective six-feature combinations with the lowest MAPE value in experiment 9. In experiment 10, there are 13, 16 and 17 test cases for each model: LSTM, GRUs and CNN. SAND will be the primary metaverse cryptocurrencies used to evaluate the predictivity of the seven-feature combinations. Five selective seven-feature combinations with the lowest MSE, MAE, MSE and MAPE values will be selected for the following experiments.

6.1.10.1 LSTM

Table 32: Results of SAND Prediction by LSTM model in Experiment 10

Feature	Metrics			
	MSE	MAE	RMSE	MAPE
Open, High, Low, Pre_Close, H-L, SMA_7, EMA_7	**8.1750**	**2.8296**	**2.8592**	**100.3319**
Open, Low, Volume, Pre_Close, H-L, SMA_7, EMA_7	15.0344	3.8648	3.8774	136.8407
Open, Low, Pre_Close, H-L, SMA_7, rsi_7, EMA_7	11.0693	3.3036	3.3271	116.9730
Open, High, Low, Volume, Pre_Close, SMA_7, EMA_7	**7.4893**	**2.6157**	**2.7367**	**92.8950**
Open, High, Low, Pre_Close, H-L, SMA_7, EMA_7	8.1750	2.8296	2.8592	100.3319
Open, High, Low, Pre_Close, 0-C, SMA_7, EMA_7	**6.8452**	**2.5170**	**2.6163**	**89.3215**
Open, High, Low, Pre_Close, SMA_7, rsi_7, EMA_7	**9.0236**	**2.2622**	**3.0039**	**80.7499**
High, Low, Volume, Pre_Close, H-L, SMA_7, EMA_7	14.8882	3.8389	3.8585	135.9472
High, Low, Volume, Pre_Close, 0-C, SMA_7, EMA_7	12.5433	3.5251	3.5417	124.8578
High, Low, Volume, Pre_Close, SMA_7, rsi_7, EMA_7	**11.6344**	**3.1697**	**3.1697**	**112.6829**
Open, High, Volume, Pre_Close, H-L, SMA_7, EMA_7	12.4992	3.4959	3.5354	123.9582
Open, High, Volume, Pre_Close, 0-C, SMA_7, EMA_7	12.0863	3.4100	3.4765	120.9489
Open, High, Volume, Pre_Close, SMA_7, rsi_7, EMA_7	11.1650	3.3105	3.3414	117.3367

Note: Bolded values indicate the 5 best results(7 features).The 5 best results features will be selected for the next LSTM experiments by adding up 1 feature.

Table 32 above summarizes the results of SAND Prediction by LSTM model in Experiment 10. In experiment 10 using LSTM as the predictive model, "Open, High, Low, Pre_Close, H-L, SMA_7, EMA_7", "Open, High, Low, Volume, Pre_Close, SMA_7 EMA_7", "Open, High, Low, Pre_Close, 0-C, SMA_7, EMA_7", "Open, High, Low, Pre_Close, SMA_7, rsi_7, EMA_7" and "High, Low, Volume, Pre_Close, SMA_7, rsi_7, EMA_7" are the top 5 four-feature combinations that have the top 5 lowest MAPE values which are recorded as 100.3319, 92.895, 89.3215, 80.7499 and 112.6829. Moreover, among these features, the "Open, High, Low, Pre_Close, SMA_7, rsi_7, EMA_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 80.7499%.

6.1.10.2 GRUS

Table 33: Results of SAND Prediction by GRUs model in Experiment 10

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Volume, Pre_Close, 0-C, H-L, rsi_7	**0.4385**	**0.5001**	**0.6622**	**17.8521**
Open, Low, Volume, Pre_Close, 0-C, H-L, rsi_7	1.4174	1.1389	1.1905	40.4278
Open, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7	1.4168	1.0466	1.1903	37.3368
Open, Volume, Pre_Close, 0-C, H-L, rsi_7, EMA_7	**0.2745**	**0.3735**	**0.5239**	**13.2921**
Open, High, Volume, Pre_Close, H-L, rsi_7, EMA_7	0.6121	0.6335	0.7824	22.1996
Open, Low, Volume, Pre_Close, H-L, rsi_7, EMA_7	0.7197	0.7018	0.8484	24.9890
Open, Volume, Pre_Close, H-L, SMA_7, rsi_7, EMA_7	1.1688	1.0154	1.0811	35.6892
Open, High, Low, Volume, H-L, 0-C, rsi_7	**0.1791**	**0.3815**	**0.4232**	**13.4033**
Open, Low, Volume, H-L, 0-C, SMA_7, rsi_7	2.4111	1.5147	1.5528	53.7941

Open, Low, Volume, H-L, 0-C, rsi_7, EMA_7	0.4277	0.6086	0.6540	21.3757
Open, High, Low, Volume, Pre_Close, 0-C, rsi_7	4.0340	1.8864	2.0085	67.1861
Open, High, Volume, Pre_Close, 0-C, SMA_7, rsi_7	0.5608	0.6174	0.7489	21.8236
Open, High, Volume, Pre_Close, 0-C, rsi_7, EMA_7	**0.4766**	**0.5682**	**0.6903**	**20.2701**
High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7	**0.4000**	**0.5299**	**0.6325**	**18.8906**
High, Low, Volume, Pre_Close, 0-C, SMA_7, rsi_7	0.5787	0.6476	0.7607	23.1336
High, Low, Volume, Pre_Close, 0-C, rsi_7, EMA_7	0.5061	0.6025	0.7114	21.5212

Note: Bolded values indicate the 5 best results(7 features).The 5 best results features will be selected for the next GRUs experiments by adding up 1 feature.

Table 33 above summarizes the results of SAND Prediction by GRUs model in Experiment 10. In experiment 10 using GRUs as the predictive model, "Open, High, Volume, Pre_Close, 0-C, H-L, rsi_7", "Open, Volume, Pre_Close, 0-C, H-L, rsi_7, EMA_7", "Open, High, Low, Volume, H-L, 0-C, rsi_7", "Open, High, Volume, Pre_Close, 0-C, rsi_7, EMA_7" and "High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7" are the top 5 seven-feature combinations that have the top 5 lowest MAPE value which is recorded as 17.8521, 13.2921, 13.4033, 20.2701 and 18.8906. Moreover, among these features, the "Open, Volume, Pre_Close, 0-C, H-L, rsi_7, EMA_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 13.2921%.

6.1.10.3 CNN

Table 34: Results of SAND Prediction by CNN model in Experiment 10

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Low, Pre, Close, 0-C, SMA_7, EMA_7	6.9084	2.1841	2.6284	77.5467
Open, High, Volume, Pre, Close, 0-C, SMA_7, EMA_7	4.6306	1.8709	2.1519	65.9172
Open, High, Pre, Close, H-L, 0-C, SMA_7, EMA_7	2.9774	1.2821	1.7255	45.0312
Open, High, Pre, Close, 0-C, SMA_7, rsi_7, EMA_7	1.7310	1.2042	1.3157	42.8696
Open, High, Low, Pre, Close, 0-C, rsi_7, EMA_7	3.3705	1.7676	1.8359	62.3772
High, Low, Volume, Pre, Close, 0-C, rsi_7, EMA_7	5.9578	2.2526	2.4409	79.5750
High, Low, Pre, Close, H-L, 0-C, rsi_7, EMA_7	3.6212	1.6806	1.9029	59.4199
High, Low, Pre, Close, 0-C, SMA_7, rsi_7, EMA_7	**0.7492**	**0.7804**	**0.8656**	**27.7710**
Open, High, Volume, Pre, Close, 0-C, rsi_7, EMA_7	**0.8051**	**0.7971**	**0.8973**	**28.2024**
Open, High, Pre, Close, H-L, 0-C, rsi_7, EMA_7	**0.4575**	**0.5231**	**0.6764**	**18.7678**
Open, High, Low, 0-C, SMA_7, rsi_7, EMA_7	1.4756	1.0329	1.2147	36.5403
Open, High, Volume, 0-C, SMA_7, rsi_7, EMA_7	**0.6914**	**0.7519**	**0.8315**	**26.5309**
Open, High, H-L, 0-C, SMA_7, rsi_7, EMA_7	1.1136	1.1136	1.0553	32.3574
Open, High, Volume, Pre, Close, SMA_7, rsi_7, EMA_7	0.8790	0.7985	0.9376	28.2060
High, Low, Volume, Pre, Close, SMA_7, rsi_7, EMA_7	2.6376	1.2369	1.6241	44.1382
High, Volume, Pre, Close, H-L, SMA_7, rsi_7, EMA_7	4.6692	1.8736	2.1608	65.5486
High, Volume, Pre, Close, 0-C, SMA_7, rsi_7,EMA_7	**0.7551**	**0.7083**	**0.8690**	**24.9789**

Note: Bolded values indicate the 5 best results(7 features).The 5 best results features will be selected for the next CNN experiments by adding up 1 feature.

Table 34 above summarizes the results of SAND Prediction by CNN model in Experiment 10. In experiment 10 using LSTM as the predictive model, High, Low, Pre, Close, 0-C, SMA_7, rsi_7, EMA_7", "Open, High, Volume, Pre, Close, 0-C, rsi_7, EMA_7", "Open, High, Pre, Close, H-L, 0-C, rsi_7, EMA_7", "Open, High, Volume, 0-C, SMA_7, rsi_7, EMA_7" and "High, Volume, Pre, Close, 0-C, SMA_7, rsi_7, EMA_7" are the top 5 seven-feature combinations that have the top 5 lowest MAPE values which are recorded as 27.771, 28.2024, 18.7678, 26.5309 and 24.9789. Moreover, among these features, the "Open, High, Pre, Close, H-L, 0-C, rsi_7, EMA_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 18.7678 %.

6.1.11 Experiment 11 (8 features)

The experiment 11 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in this section implicitly. Experiment 11 uses eight dependent features as input to predict the next ten days of closing price, labelled as "Close" target features. Experiment 11 evaluates the predictability of eight-feature combinations by adding one more feature individually to the five selective seven-feature combinations with the lowest MAPE value in experiment 10. In experiment 11, there are 8 test cases for LSTM and 12 test cases for GRUs and CNN.SAND will be the primary metaverse cryptocurrencies used to evaluate the predictivity of the eight-feature combinations. Five selective eight-feature combinations with the lowest MSE, MAE, MSE and MAPE values will be selected for the following experiments.

6.1.11.1 LSTM

Table 35: Results of SAND Prediction by LSTM model in Experiment 11

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Low, Volume, Pre_Close, SMA_7, rsi_7, EMA_7	**7.4583**	**2.2624**	**2.7310**	**80.6640**
Open, High, Low, Pre_Close, H-L, SMA_7, rsi_7, EMA_7	**10.4619**	**3.1590**	**3.2345**	**112.0878**
Open, High, Low, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7	**8.9536**	**2.8650**	**2.9923**	**101.6109**
Open, High, Low, Volume, Pre_Close, 0-C, SMA_7, EMA_7	**11.2863**	**3.3066**	**3.3595**	**117.2192**
Open, High, Low, Pre_Close, H-L, 0-C, SMA_7, EMA_7	**8.3113**	**2.8640**	**2.8829**	**101.4039**
Open, High, Low, Volume, Pre_Close, H-L, SMA_7, EMA_7	12.6552	3.5288	3.5574	125.0410
High, Low, Volume, Pre_Close, H-L, SMA_7, rsi_7, EMA_7	11.4594	3.3730	3.3852	119.4046
High, Low, Volume, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7	13.5153	3.5652	3.6763	126.4328

Note: Bolded values indicate the 5 best results(8 features).The 5 best results features will be selected for the next LSTM experiments by adding up 1 feature.

Table 35 above summarizes the results of SAND Prediction by LSTM model in Experiment 11. In experiment 11 using LSTM as the predictive model, "Open, High, Low, Volume, Pre_Close, SMA_7, rsi_7, EMA_7", "Open, High, Low, Pre_Close, H-L, SMA_7, rsi_7, EMA_7", "Open, High, Low, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7", "Open, High, Low, Volume, Pre_Close, 0-C, SMA_7, EMA_7" and "Open, High, Low, Pre_Close, H-L, 0-C, SMA_7, EMA_7" are the top 5 eight-feature combinations that have the top 5 lowest MAPE values which are recorded as 80.664, 112.0878, 101.6109, 117.2192 and 101.4039. Moreover, among these features, the "Open, High, Low, Volume, Pre_Close, SMA_7, rsi_7, EMA_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 80.664 %.

6.1.11.2 GRUs

Table 36: Results of SAND Prediction by GRUs model in Experiment 11

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7	1.3700	0.9485	1.1705	33.7788
Open, Low, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7	4.9281	2.1146	2.2199	75.1659
Open, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7	1.9591	1.3778	1.3997	48.6546
Open, High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7	**0.2007**	**0.3955**	**0.4480**	**13.8631**
Open, High, Low, Volume, H-L, 0-C, SMA_7, rsi_7	**0.6876**	**0.7345**	**0.8292**	**26.1901**
Open, High, Low, Volume, H-L, 0-C, rsi_7, EMA_7	**0.4989**	**0.6196**	**0.7063**	**21.7326**
Open, High, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7	**0.6609**	**0.6527**	**0.8130**	**23.2986**
Open, High, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7	1.3700	0.9485	1.1705	33.7788
High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7	1.1109	0.9375	1.0540	33.4820
High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7	**0.8804**	**0.7568**	**0.9383**	**26.9836**
Open, High, Low, Volume, Pre_Close, 0-C, rsi_7, EMA_7	0.8663	0.7685	0.9307	27.0105
Open, High, Volume, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7	1.0149	0.8400	1.0074	29.5943

Note: Bolded values indicate the 5 best results(8 features).The 5 best results features will be selected for the next GRUs experiments by adding up 1 feature.

Table 36 above summarizes the results of SAND Prediction by GRUs model in Experiment 11. In experiment 11 using GRUs as the predictive model, "Open, High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7", "Open, High, Low, Volume, H-L, 0-C, SMA_7, rsi_7", "Open, High, Low, Volume, H-L, 0-C, rsi_7, EMA_7", "Open, High, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7" and "High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7" are the top 5 eight-feature combinations that have the top 5 lowest MAPE values which are recorded as 13.8631, 26.1901, 21.7326, 23.2986 and 26.9836. Moreover, among these features, the "Open, High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 13.8631%.

6.1.11.3 CNN

Table 37: Results of SAND Prediction by CNN model in Experiment 11

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Low, Pre_Close, H-L, 0-C, rsi_7, EMA_7	2.4978	1.2807	1.5804	45.3912
Open, High, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7	**1.2034**	**0.9934**	**1.0970**	**35.0034**
Open, High, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7	10.1551	2.7928	3.1867	99.0854
Open, High, Volume, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7	4.7527	1.8139	2.1801	64.0612
High, Low, Volume, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7	**0.7632**	**0.8200**	**0.8736**	**28.7848**
High, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7	3.6695	1.4371	1.9156	51.4164
Open, High, Low, Volume, 0-C, SMA_7, rsi_7, EMA_7	5.0476	1.9534	2.2467	69.7517
Open, High, Volume, H-L, 0-C, SMA_7, rsi_7, EMA_7	**1.8041**	**1.0596**	**1.3432**	**37.4777**
Open, High, Low, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7	1.8873	1.1447	1.3738	40.0793
High, Low, Volume, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7	**0.7607**	**0.7362**	**0.8722**	**25.8021**
High, Low, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7	2.8581	1.4164	1.6906	50.3032
Open, High, Low, Volume, Pre_Close, 0-C, rsi_7, EMA_7	**1.5578**	**1.0266**	**1.2481**	**36.2353**

Note: Bolded values indicate the 5 best results(8 features).The 5 best results features will be selected for the next CNN experiments by adding up 1 feature.

Table 37 above summarizes the results of SAND Prediction by CNN model in Experiment 11. In experiment 11 using CNN as the predictive model, "Open, High, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7,"High, Low, Volume, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7", "Open, High, Volume, H-L, 0-C, SMA_7, rsi_7, EMA_7", "High, Low, Volume, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7", and "Open, High, Low, Volume, Pre_Close, 0-C, rsi_7, EMA_7" are the top 5 eight-feature combinations that have the top 5 lowest MAPE values which are recorded as 35.0034, 28.7848, 37.4777, 25.8021 and 36.2353. And among these features, the "High, Low, Volume, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 25.8021%.

6.1.12 Experiment 12 (9 features)

The experiment 12 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in this section implicitly. Experiment 12 uses nine dependent features as input to predict the next ten days of closing price, labelled as "Close" target features. Experiment 12 evaluates the predictability of nine-feature combinations by adding one more feature individually to the five selective eight-feature combinations with the lowest MAPE value in experiment 11. In experiment 12, there are 6 test cases for LSTM, GRUs and CNN. SAND will be the primary metaverse cryptocurrencies used to evaluate the predictivity of the nine-feature combinations. Five selective nine-feature combinations with the lowest MSE, MAE, MSE and MAPE values will be selected for the following experiments.

6.1.12.1 LSTM

Table 38: Results of SAND Prediction by LSTM model in Experiment 12

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Low, Volume, Pre_Close, H-L, SMA_7, rsi_7, EMA_7	**11.6341**	**3.3750**	**3.4109**	**119.6881**
Open, High, Low, Volume, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7	**13.9866**	**3.6057**	**3.7399**	**128.0813**
Open, High, Low, Pre_Close, 0-C, H-L, SMA_7, rsi_7, EMA_7	**12.4813**	**3.5164**	**3.5329**	**124.6004**
Open, High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, EMA_7	**15.2133**	**3.8874**	**3.9004**	**137.5815**
High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7	**14.9281**	**3.8490**	**3.8637**	**136.2467**

Note: Bolded values indicate the 5 best results(9 features).The 5 best results features will be selected for the next LSTM experiments by adding up 1 feature.

Table 38 above summarizes the results of SAND Prediction by LSTM model in Experiment 12. In experiment 12 using LSTM as the predictive model, "Open, High, Low, Volume, Pre_Close, H-L, SMA_7, rsi_7, EMA_7", "Open, High, Low, Volume, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7", "Open, High, Low, Pre_Close, 0-C, H-L, SMA_7, rsi_7, EMA_7", "Open, High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, EMA_7", and "High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7" are the top 5 nine-feature combinations that have the top 5 lowest MAPE values which are recorded as 119.6881, 128.0813, 124.6004, 137.5815 and 136.2467. Moreover, among these features, the "Open, High, Low, Volume, Pre_Close, H-L, SMA_7, rsi_7, EMA_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 119.6881 %.

6.1.12.2 GRUs

Table 39: Results of SAND Prediction by GRUs model in Experiment 12

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7	**0.1588**	**0.3511**	**0.3986**	**12.3334**
Open, High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7	**0.3048**	**0.4982**	**0.5520**	**17.4664**
Open, High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7	4.3463	1.9283	2.0848	68.6456
Open, High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7	**2.4538**	**1.5039**	**1.5665**	**53.4654**
High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7	**2.8515**	**1.4203**	**1.6886**	**50.7487**
Open, High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7	**1.6928**	**1.2439**	**1.3011**	**44.2754**

Note: Bolded values indicate the 5 best results(9 features).The 5 best results features will be selected for the next LSTM experiments by adding up 1 feature.

Table 39 above summarizes the results of SAND Prediction by GRUs model in Experiment 12. In experiment 12 using GRUs as the predictive model, "Open, High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7", "Open, High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7", "Open, High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7", "High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7", "Open, High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7" are the top 5 nine-feature combinations that have the top 5 lowest MAPE values which are recorded as 12.3334, 17.4664, 53.4654, 50.7487 and 44.2754. Moreover, among these features, the "Open, High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 12.3334%.

6.1.12.3 CNN

Table 40: Results of SAND Prediction by CNN model in Experiment 12

Feature	Metrics			
	MSE	MAE	MSE	MAPE
Open, High, Low, Volume, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7	**1.2366**	**0.9570**	**1.1120**	**33.9509**
Open, High, Low, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7	3.6685	1.5714	1.9153	55.7538
High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7	**0.3605**	**0.5177**	**0.6004**	**18.2743**
Open, High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7	**1.5594**	**1.0766**	**1.2487**	**37.9821**
Open, High, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7	**2.9993**	**1.4682**	**1.7319**	**52.0553**
Open, High, Low, Volume, H-L, 0-C, SMA_7, rsi_7, EMA_7	**2.2819**	**1.1365**	**1.5106**	**40.1479**

Note: Bolded values indicate the 5 best results(9 features).The 5 best results features will be selected for the next LSTM experiments by adding up 1 feature.

Table 40 above summarizes the results of SAND Prediction by CNN model in Experiment 12. In experiment 12 using CNN as the predictive model, "Open, High, Low, Volume, Pre_Close, 0-C, SMA_7, rsi_7, EMA_7", "High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7", "Open, High, Low, Volume, Pre_Close, H-L, 0-C, rsi_7, EMA_7", "Open, High, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7", "Open, High, Low, Volume, H-L, 0-C, SMA_7, rsi_7, EMA_7" are the top 5 nine-feature combinations that have the top 5 lowest MAPE values which are recorded as 33.9509, 18.2743, 37.9821, 52.0553 and 40.1479. Moreover, among these features, the "High, Low, Volume, Pre_Close, H-L, 0-C, SMA_7, rsi_7, EMA_7" have the highest predictive power as the lowest average of the absolute percentage errors of forecasts is recorded as 18.2743 %.

6.1.13 Experiment 13 (All features)

The experiment 13 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in the after sections implicitly. Experiment 13 uses all features in the curated dataset, which are Open, High, Low, Volume, Pre, Close, H-L, 0-C, SMA_7, rsi_7, and EMA_7 as input features to predict the next ten days of closing price, labelled as 'Close' target features.

Table 41: Results of SAND Prediction by both 3 models in Experiment 13

Model	Metrics			
	MSE	MAE	RMSE	MAPE
LSTM	12.9718	3.5766	3.6016	126.7566
GRUs	0.1328	0.3083	0.3645	10.8018
CNN	0.4098	0.388	0.6402	13.8996

Table 41 above summarizes the results of SAND Prediction of both 3 models in Experiment 13. In experiment 13, as LSTM as predictive model, MAPE has been recorded as 126.7566 to predict the closing price of the SAND. The other models, CNN and LSTM, are recorded as means absolute percentage errors of forecasts of 10.8018% and 13.8996%.

6.2 Results for Feature Selection Experiments (1-13)

Table 42: Objective and Features for Experiments 1-13

Experiments	Experiments Objective	Features
1	Test Predictivity of The Previous Closing Price(PC)	Pre_Close
2	Test Predictivity of The Primary Technical Indicator(PTI)	Open, High, Low, Volume
3	Test Predictivity of The Advanced Technical Indicator(ATI)	H-L, 0-C, SMA_7, rsi_7, EMA_7
4-13	Find the Best Optimal Features(BOF)	LSTM: Volume GRUs: Open, Low, Volume, H-L, 0-C, rsi_7 CNN: rsi_7, EMA_7

Table 42 above summarizes the experimental objective and features for experiments 1 to 13. In experiments 1, the experiments objective is testing the predictivity of the previous closing price(PC) to test the "Pre_Close" features in the SAND datasets. In experiments 2, the experiments objective is testing the primary technical indicator (PTI) to test the "Open, High, Low, Volume" features in the SAND datasets. In experiments 3, the experiments objective is testing the advanced technical indicator (ATI) to test the "Pre_Close, H-L, 0-C, SMA_7, rsi_7" features in the SAND, SLP and MANA datasets. In experiments 4 to 13 is testing the predictivity of 1 feature to 10 feature as the input features. After conducted various experiments the features selected for Best Optimal Features(BOF) is "Volume" for LSTM, "Open, Low, Volume, H-L, 0-C, rsi_7" for GRUs and "rsi_7, EMA_7" for CNN.

Table 43: Overall Result for Experiments 1 to 13

Currencies	Model	Experiments	Metrics			
			MSE	MAE	MSE	MAPE
Sand	LSTM	PC	5.5697	2.3422	2.3600	82.9270
		PTI	10.3318	3.1820	3.2143	112.8267
		ATI	1.0109	3.6440	3.6498	128.9782
		BOF	**0.2286**	**0.3776**	**0.4781**	**13.3211**
	Grus	PC	0.8260	0.7819	0.9088	27.9069
		PTI	0.8384	0.7150	0.9156	25.2571
		ATI	1.9047	0.8952	1.0054	31.2915
		BOF	**0.0795**	**0.2443**	**0.2820**	**8.5567**
	CNN	PC	1.3917	0.9503	1.1797	33.6532
		PTI	6.6546	2.1983	2.5796	77.3749
		ATI	7.2024	2.4872	2.6837	88.2758
		BOF	**0.0617**	**0.1974**	**0.2483**	**6.9224**
SLP	LSTM	PC	0.0043	0.0627	0.0657	351.9648
		PTI	**0.0013**	**0.0353**	**0.0354**	**200.9148**
		ATI	0.0072	0.0845	0.0851	478.2201
		BOF	0.0015	0.0389	0.0389	221.2538
	Grus	PC	**0.0019**	**0.0419**	**0.0441**	**236.1956**
		PTI	0.0278	0.1628	0.1668	933.8402
		ATI	0.0042	0.0631	0.0645	356.3695
		BOF	0.0140	0.1175	0.1185	664.4986
	CNN	PC	**0.0006**	**0.0219**	**0.0246**	**122.9813**
		PTI	0.0008	0.0243	0.0291	140.4431
		ATI	0.0180	0.1211	0.1342	674.6060
		BOF	0.0180	0.1405	0.1411	798.1401
MANA	LSTM	PC	0.8891	0.9393	0.9429	42.9516
		PTI	**0.1563**	**0.3839**	**0.3953**	**17.5064**
		ATI	2.1293	1.4565	1.4592	66.6537
		BOF	4.0399	2.0054	2.0099	91.8533
	Grus	PC	1.0526	1.0220	1.0260	46.7369
		PTI	2.5632	1.5957	1.6010	73.0209
		ATI	4.1786	2.0419	2.0442	93.4819
		BOF	**0.0629**	**0.2194**	**0.2508**	**10.0421**
	CNN	PC	**0.0690**	**0.2517**	**0.2628**	**11.4615**
		PTI	1.6406	1.2563	1.2809	57.4322
		ATI	0.5524	0.7069	0.7433	32.2474
		BOF	0.8459	0.9133	0.9197	41.7894

Note: Bolded values indicate the best results.

Table 43 above summarizes overall result for experiments 1 to 13. Experiments using LSTM, GRU, and CNN models to predict the closing price of SAND cryptocurrency using PC, PTI, ATI, and BOF features have recorded MAPE values of 82.4799, 112.8267 ,128.9782 and 13.3211(LSTM), 27.9069, 25.2571 and 31.2915 and 8.5567(GRUs) and 33.6532, 77.3749, 88.2758 and 6.9224(CNN).

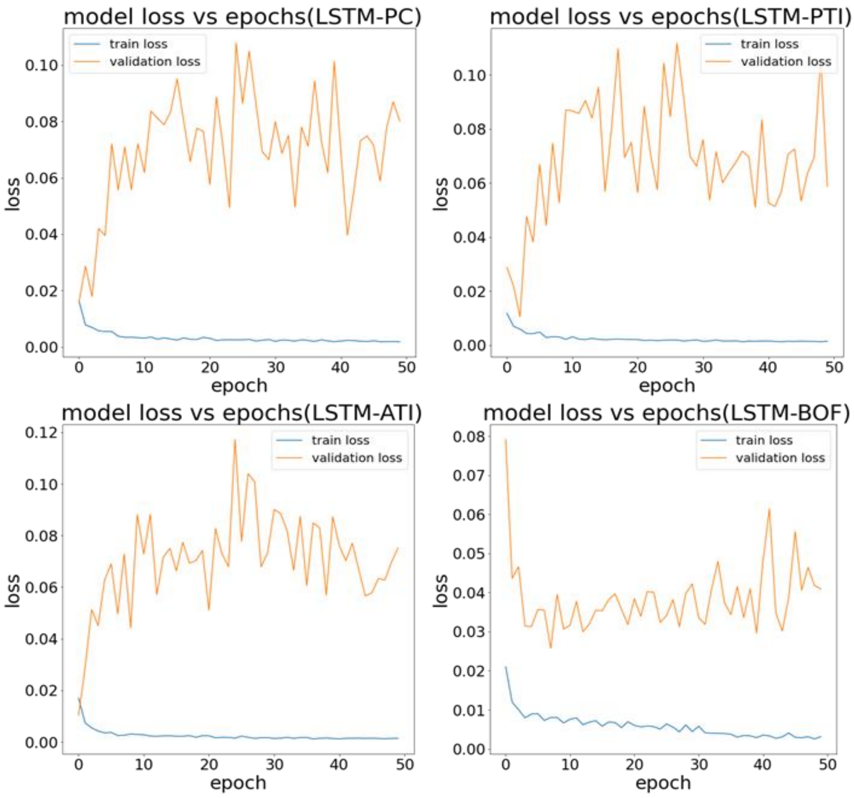
The BOF feature consistently results in the lowest mean absolute percentage error (MAPE) value when using the LSTM, GRU, and CNN models when predicting the SAND. This suggests that the BOF feature may be a strong predictor of the closing price of SAND cryptocurrency. The experiments will be repeated by using the SLP and MANA as other test cases.

Experiments using LSTM, GRU, and CNN models to predict the closing price of SLP as the test case to test the compatibility of the SAND models to another cryptocurrencies model. By using PC, PTI, ATI, and BOF features, the results have recorded MAPE values of 351.9648, 200.9148, 478.2201 and 221.2538 (LSTM), 236.1956, 933.8402 and 356.3695 and 664.4986(GRUs) and 122.9813, 140.4431, 674.6060 and 798.1401 (CNN). This indicating that the models have a low accuracy in predicting the closing price of SLP. This suggests that either these features are not good predictors, or the model and features used is no suitable for the SLP dataset. Further investigation is needed to identify the cause and improve the prediction accuracy.

Experiments using LSTM, GRU, and CNN models to predict the closing price of MANA as another test case to test the compatibility of the SAND models to another cryptocurrencies model. By using PC, PTI, ATI, and BOF features, the results have recorded MAPE values of 42.9516, 17.5064, 66.6537 and 91.8533 (LSTM), 46.7369, 73.0209,93.4819 and 10.0421(GRUs) and 11.4615, 57.4322, 32.2474 and 41.7894 (CNN). The results have recorded a range of MAPE values, with some being relatively low and others being relatively high. This suggests that the models that were trained on the SAND cryptocurrency may not be as compatible with the MANA cryptocurrency, and that further adjustments or retraining may be necessary to improve the prediction accuracy for MANA. It's also possible that the features used in these experiments may not be as informative for the MANA cryptocurrency as they were for SAND.

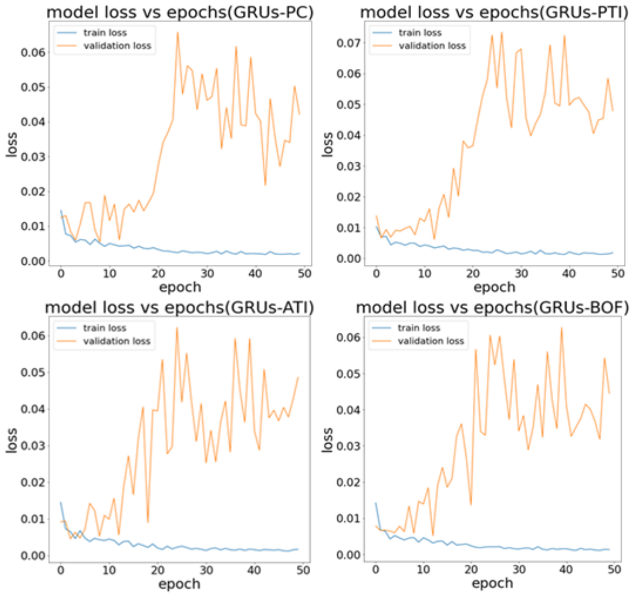
6.3 Discussion for Feature Selection Experiments (1-13)

Figure 69:Model loss vs Epochs(LSTM)for SAND



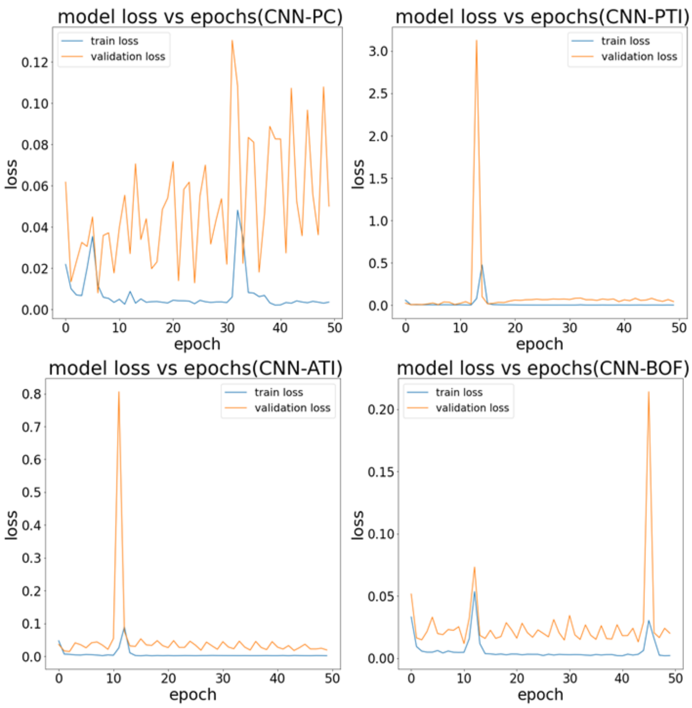
From Figure 69, the train loss is decrease exponentially for the LSTM models to predict SAND by using the PC, PTI and ATI feature, where the validation loss is increase over the epochs. This happened when a model is overfitting, which means it has learned the training data too well and performs poorly on new, unseen data. The models have memorized the training data and is not generalizing well to the validation data. By using the BOF feature, the train and validation loss decrease exponentially, until the certain epochs then the validation loss increase .This means the model is initially learning the patterns in the training data and generalizing well to the validation data, but as it continues to train, it starts to memorize the training data rather than generalizing to new unseen data. This causes the validation loss to increase, indicating that the model is not performing well on new data.

Figure 70: Model loss vs Epochs(GRUs)for SAND



From Figure 70, the train loss is decrease exponentially for the LSTM models by using the PC, PTI and ATI feature, where the validation loss is increase over the epochs. This happened when a model is overfitting, which means it has learned the training data too well and performs poorly on new, unseen data. The models have memorized the training data and is not generalizing well to the validation data.

Figure 71: Model loss vs Epochs(CNN)for SAND



From Figure 71, a train loss curve with two spikes is observed. That can be occur due to several reasons which are learning rate, optimizer, and the model architecture. For learning rate that is initially high and then reduced after a certain number of epochs, the model may make large parameter updates in the beginning, causing the loss to spike. That can be caused also by the model architectures used that take longer to converge causing the loss to spike.

From the results gained at 3 figures above, it seems most of the models face the problem of the overfitting. There are several factors that causing of the overfitting of the model that are will be explained implicitly in the Table 44 below.

Table 44: Factor of Overfitting and Ways to Hyper Tune

Factor of overfitting	Explanation	Ways To Hyper tune
High model complexity	High model complexity cause the models learn the noise in the training data, rather than the underlying pattern, which can lead to overfitting.	The change of the architecture of the model and the different complexity is test through the experiments instead of the default complexity.
Lack of regularization	Regularization can prevent overfitting by prevent the model too relies to any one feature by adding penalty term to loss function. L1 Regularization(ridge regression) adds the "absolute value of magnitude" of the coefficient as the penalty term to the loss function. L2 Regularization (ridge regression) adds the "squared magnitude" of the coefficient as the penalty term to the loss function. Dropout Regularization randomly drop out a certain percentage of neurons during training of the models.	The L1 and L2 Regularization will be added into the architecture of the LSTM, GRUs and CNN models.
High training epochs	The high training epochs will cause model more specialized to the training data. Models perform well on the training data but poorly on new data.	The different epochs is test through the experiments instead of the default epochs.
High batch size	Batch size is the number of samples used in one iteration of the training process. The large batch size will cause the model may converge too quickly and overfit to the training data.	The different batch is test through the experiments instead of the default batch size.
High learning rate	The learning rate is a hyperparameter that controls how fast a model learns from the training data. High learning rate can cause overfitting because the model may learn too quickly and overshoot the optimal solution	The different learning rate is test through the experiments instead of the default learning rate.

6.4 Hyperparameter Tuning Experiments (14-20)

This section discusses the results of experiments 14 to 20 to three metaverse cryptocurrencies, SAND, SLP and MANA, in the metrics of MSE, MAE, RMSE, and MAPE. SAND will be the primary cryptocurrencies, and the SLP and MANA will be the secondary cryptocurrencies used for the test cases when select the SAND best parameter. With a comparative analysis of the value of the MAPE in these experiments, the best model and the feature used can be determined for these three metaverse cryptocurrencies.

6.4.1 Experiment 14 (Learning rate)

The experiment 14 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in the after sections implicitly. In experiment 14, the next 10 days closing price of a metaverse cryptocurrency was predicted using LSTM, GRUs, and CNN models with input features Pre_close(PC), Primary-technical indicator(PTI), Advance technical indicator(ATI), Best Optimal Features(BOF). Hyperparameter tuning was conducted by testing different learning rate values (0.00001, 0.0001, 0.0005, 0.01) in place of the default value of 0. 001, and the learning rate with the lowest MAPE value was chosen as the setting for future experiments because it resulted in the most accurate predictions according to the MAPE metric, which measures the average absolute percentage error of the model's predictions.

6.4.1.1 LSTM

Table 45: Results of SAND Prediction by LSTM model in Experiment 14

Test Case	Learning Rate	Metrics			
		MSE	MAE	RMSE	MAPE
Pre_Close(PC)	0.00001	**2.8164**	**1.6609**	**1.6782**	**58.8019**
	0.0001	4.4231	2.0697	2.1031	73.3577
	0.0005	5.4312	2.2879	2.3305	81.0758
	0.001(default)	5.5697	2.3422	2.3600	82.9270
	0.01	3.8516	1.8087	1.9625	64.3439
Primary Technical Indicator(PTI)	0.00001	3.6324	1.8940	1.9059	67.0751
	0.0001	10.3537	3.1036	3.2177	110.1659
	0.0005	9.1550	2.8332	3.0257	100.6498
	0.001(default)	10.3318	3.1820	3.2143	112.8267
	0.01	**3.0280**	**1.7366**	**1.7401**	**61.2700**

Advance Technical Indicator(ATI)	0.00001	4.2135	2.0372	2.0527	72.1118
	0.0001	8.6285	2.9173	2.9374	103.0840
	0.0005	9.8167	3.1302	3.1332	110.6758
	0.001(default)	13.3196	3.6438	3.6496	128.9707
	0.01	**4.0557**	**1.9795**	**2.0139**	**69.8294**
Best Optimal Features(BOF)	0.00001	10.0070	3.0981	3.1634	109.5097
	0.0001	3.3342	1.8215	1.8260	64.3421
	0.0005	2.3836	1.5389	1.5439	54.4854
	0.001(default)	**0.2286**	**0.3776**	**0.4782**	**13.3211**
	0.01	0.3418	0.4942	0.5846	17.2431

Note: Bolded values indicate the best results.
The best results were obtained for 4 test cases for learning rate. This information is used in Experiment 15(LSTM model) to test different batch size.

From Table 45, it appears that as both three feature (PC, PTI and ATI) undergo the logarithmic growth in the values of MAPE when the table until the learning rate of 0.001 and then show the gradual decrease as the learning rate increases. The best learning rate for PC is 0.00001 with lowest MAPE of 58.8019. The best learning rate for PTI is 0.01 with lowest MAPE of 61.2700. The best learning rate for ATI is 0.01 with lowest MAPE of 69.8294. For the BOF, the plot graph between the learning rate value as X axis and MAPE values as Y-axis in like the training-loss curve (decrease rapidly and then level off to a low value and increase after certain point), the value of the 13.3211 is the learning rate that recorded with the lowest MAPE value.

6.4.1.2 GRUs

Table 46: Results of SAND Prediction by GRUs model in Experiment 14

Test Case	Learning Rate	Metrics			
		MSE	MAE	MSE	MAPE
Pre_Close(PC)	0.00001	**0.2853**	**0.5038**	**0.5341**	**17.9180**
	0.0001	0.3113	0.5124	0.5579	18.2179
	0.0005	0.8861	0.7945	0.9414	28.3410
	0.001(default)	0.8259	0.7819	0.9088	27.9049
	0.01	3.1131	1.7521	1.7644	62.0857
Primary Technical Indicator(PTI)	0.00001	**0.1004**	**0.2546**	**0.3169**	**9.1135**
	0.0001	0.7573	0.8581	0.8702	30.2534
	0.0005	0.9467	0.8476	0.9730	29.7844
	0.001(default)	0.8383	0.7149	0.9156	25.2571
	0.01	2.8642	1.6889	1.6924	59.5847
Advance Technical Indicator(ATI)	0.00001	**0.1903**	**0.3278**	**0.4362**	**11.7542**
	0.0001	1.4918	1.2030	1.2214	42.6537
	0.0005	9.6907	2.8614	3.1130	101.8590
	0.001(default)	1.0043	0.8929	1.0021	31.2915
	0.01	6.9861	2.6325	2.6431	93.2169

	0.00001	1.8320	1.3253	1.3535	47.0041
Best	0.0001	3.3471	1.8146	1.8295	64.2987
Optimal	0.0005	13.6625	3.5843	3.6963	127.2662
Features(BOF)	0.001(default)	**0.0795**	**0.2443**	**0.2820**	**8.5567**
	0.01	1.5587	1.2352	1.2485	43.6565

Note: Bolded values indicate the best results.
The best results were obtained for 4 test cases. This information is used in Experiment 15(GRUs model) to test different batch size.

From the Table 46 above, it appears that as both four feature (PC,PTI,ATI and BOF) show the increase in the values of MAPE when until the learning rate of 0.005 and then show the decrease at the learning rate of 0.0005 after that show the increase at the learning rate of 0.01. The best learning rate for PC is 0.00001 with lowest MAPE of 17.9180. The best learning rate for PTI is 0.00001 with lowest MAPE of 9.1135.The degree of the rate of change of MAPE value is steeper in ATI and BOF features. The best learning rate for ATI is 0.00001 with lowest MAPE of 11.7542. The best learning rate for BOF is 0.001 with lowest MAPE of 8.5567.

6.4.1.3 CNN

Table 47: Results of SAND Prediction by CNN model in Experiment 14

Test Case	Learning Rate	Metrics			
		MSE	MAE	MSE	MAPE
	0.00001	**0.2227**	**0.3724**	**0.4719**	**13.1792**
	0.0001	4.4065	1.9162	2.0992	68.1343
Pre_Close(PC)	0.0005	2.1148	1.1655	1.4542	41.5688
	0.001(default)	1.3917	0.9503	1.1797	33.6532
	0.01	12.7083	3.3089	3.5649	117.1887
	0.00001	**2.1280**	**1.1056**	**1.4588**	**39.3208**
Primary	0.0001	2.0074	1.2331	1.4168	43.2555
Technical	0.0005	2.8108	1.2369	1.6765	43.7093
Indicator(PTI)	0.001	6.6546	2.1983	2.5796	77.3749
	0.01	3.7503	1.9249	1.9366	67.8768
	0.00001	3.6942	1.7482	1.9220	61.7991
Advance	0.0001	3.5468	1.6480	1.8833	58.8533
Technical	0.0005	**1.7343**	**0.9950**	**1.3169**	**35.6044**
Indicator(ATI)	0.001	6.9222	2.4607	2.6310	87.2851
	0.01	29.1733	4.8988	5.4012	174.5623
	0.00001	0.7838	0.7234	0.8853	25.2664
Best	0.0001	4.2951	1.8143	2.0725	64.5650
Optimal	0.0005	0.8105	0.7725	0.9003	27.1710
Features(BOF)	0.001(default)	**0.0617**	**0.1974**	**0.2483**	**6.9224**
	0.01	2.4525	1.5511	1.5660	54.7254

Note: Bolded values indicate the best results. The best results were obtained for 4 test cases. This information is used in Experiment 15(CNN model) to test different batch size.

From Table 47 above, it appears that as four features show the different trend of change of the values of MAPE as the learning rate increases. The best learning rate for PC is 0.00001 with lowest MAPE of 13.1792. The best learning rate for PTI is 0.00001 with lowest MAPE of 39.3208. The best learning rate for ATI is 0.0005 with lowest MAPE of 35.6044. The best learning rate for BOF is 0.001 with lowest MAPE of 6.9224.

6.4.1.4 Discussion of Experiment 14

Table 48: Results for best parameter in Exp 14

Model	Features	MAPE Exp 13	MAPE Exp 14	Rate Of Change (%)	Parameter
LSTM	Pre_close (PC)	82.4799	58.8019	28.71	Lr:0.00001
	Primary technical indicator (PTI)	112.8267	61.2700	45.70	Lr:0.01
	Advance technical indicator (ATI)	128.9782	69.8294	45.86	Lr:0.01
	Best optimal feature (BOF)	13.3211	**13.3211**	0.00	Lr:0.001
GRUs	Pre_close (PC)	27.9069	17.9180	35.79	Lr:0.00001
	Primary technical indicator (PTI)	25.2571	9.1135	63.92	Lr:0.00001
	Advance technical indicator (ATI)	31.2915	11.7542	62.44	Lr:0.00001
	Best optimal feature (BOF)	8.5567	**8.5567**	0.00	Lr:0.001
CNN	Pre_close (PC)	33.6532	13.1792	60.84	Lr:0.00001
	Primary technical indicator (PTI)	77.3749	39.3208	49.18	Lr:0.00001
	Advance technical indicator (ATI)	87.2851	35.6044	59.21	Lr:0.0005
	Best optimal feature (BOF)	6.9224	**6.9224**	0.00	Lr:0.001

Note: Bolded values indicate the best results.

From the Table 48, it appears that as the Both three models with the Best optimal feature archived the lowest MAPE value with the 13.3211,8.5567 and 6.9224 for both 3 models and they does no undergo rate of change from Exp. 13 to Exp. 14 which means the default values is learning rate (0.001) is used. For LSTM model with the feature of PC, PTI and ATI, the MAPE values is decrease to 58.8019, 61.2700, 69.8294 with the rate of change of 28.71,45.70,45.86 percent. For GRUs model with the feature of PC, PTI and ATI, the MAPE values is decrease to 17.9180, 9.1135, 11.7542 with the rate of change of 35.79, 63.92,62.44 percent. For CNN model with the feature of PC, PTI and ATI, the MAPE values is decrease to 13.1792, 39.3208, 35.6044 with the rate of change of 60.84, 49.18,59.21 percent.

6.4.2 Experiment 15 (Batch size)

The experiment 15 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in the after sections implicitly. In experiment 15, the next 10 days closing price of a metaverse cryptocurrency was predicted using LSTM, GRUs, and CNN models with input features including Pre_close(PC), Primary-technical indicator(PTI), Advance technical indicator(ATI), Best Optimal Features(BOF). Hyperparameter tuning was conducted by testing different batch size (8,16,64,128) in place of the default value of 32, and the batch size with the lowest MAPE value was chosen as the setting for future experiments because it resulted in the most accurate predictions according to the MAPE metric, which measures the average absolute percentage error of the model's predictions.

6.4.2.1 LSTM

Table 49: Results of SAND Prediction by LSTM model in Experiment 15

Test Case	Batch Size	Metrics			
		MSE	MAE	RMSE	MAPE
Pre_Close(PC)	8	2.3884	1.5108	1.5455	53.4877
	16	**2.1596**	**1.4468**	**1.4696**	**51.2340**
	32(default)	2.8164	1.6609	1.6782	58.8019
	64	3.2808	1.8036	1.8113	63.8534
	128	5.8664	2.4106	2.4221	85.3765
Primary Technical Indicator(PTI)	8	5.6726	2.3757	2.3817	83.8326
	16	4.0517	2.0101	2.0129	70.9404
	32(default)	3.0280	1.7366	1.7401	61.2700
	64	2.6772	1.6333	1.6362	57.6245
	128	**2.6407**	**1.2483**	**1.6250**	**44.5605**
Advance Technical Indicator(ATI)	8	2.8731	1.6911	1.6950	59.6595
	16	4.0382	2.0068	2.0095	70.8212
	32(default)	4.0557	1.9795	2.0139	69.8294
	64	5.4080	2.3229	2.3255	82.2158
	128	**1.6108**	**1.1428**	**1.2692**	**40.4135**
Best Optimal Features(BOF)	8	3.4771	1.8617	1.8647	65.8563
	16	9.5811	3.0681	3.0953	108.3080
	32(default)	0.2286	0.3776	0.4782	13.3211
	64	0.8982	0.8813	0.9477	31.3339
	128	**0.2710**	**0.3705**	**0.5206**	**13.1830**

Note: Bolded values indicate the best results.
The best results were obtained for 4 test cases for batch size. This information is used in Experiment 16(LSTM model) to test different complexity.

From Table 49, it appears that as four features show the different trend of change of the values of MAPE as the batch size increases. The best batch size for PC is 16 with lowest MAPE of 51.2340. The best batch size for PTI is 128 with lowest MAPE of 44.5605. The best batch size of ATI is 128 with lowest MAPE of 40.4135. The best batch size for BOF is 128 with lowest MAPE of 13.1830.

6.4.2.2 GRUs

Table 50: Results of SAND Prediction by GRUs model in Experiment 15

Test Case	Batch Size	Metrics			
		MSE	MAE	MSE	MAPE
Pre_Close(PC)	8	0.2551	0.4591	0.5050	16.3329
	16	0.2474	0.4615	0.4974	16.4279
	32(default)	0.2853	0.5038	0.5341	17.9180
	64	**0.2011**	**0.4306**	**0.4484**	**15.3210**
	128	0.2924	0.5215	0.5407	18.5667
Primary Technical Indicator(PTI)	8	0.1147	0.2860	0.3387	10.2372
	16	0.1009	0.2583	0.3176	9.2579
	32(default)	0.1004	0.2546	0.3169	9.1135
	64	**0.0586**	**0.1764**	**0.2421**	**6.3238**
	128	0.0711	0.1836	0.2667	6.6057
Advance Technical Indicator(ATI)	8	0.1360	0.2650	0.3688	9.5046
	16	**0.1343**	**0.2470**	**0.3665**	**8.8777**
	32(default)	0.1903	0.3278	0.4362	11.7542
	64	0.2330	0.3844	0.4827	13.7244
	128	0.2807	0.4361	0.5298	15.5744
Best Optimal Features(BOF)	8	1.2262	1.0041	1.1074	35.7986
	16	3.1730	1.7058	1.7813	60.5916
	32(default)	**0.0795**	**0.2443**	**0.2820**	**8.5567**
	64	0.4301	0.5306	0.6558	18.8202
	128	5.4554	2.1535	2.3357	76.6796

Note: Bolded values indicate the best results.
The best results were obtained for 4 test cases for batch size. This information is used in Experiment 16(GRUs model) to test different complexity.

From Table 50, it appears that as four features show the different trend of change of the values of MAPE as the batch size increases. The best batch size for PC is 64 with lowest MAPE of 15.3210. The best batch size for PTI is 64 with lowest MAPE of 6.3238. The best batch size of ATI is 16 with lowest MAPE of 8.8777. The best batch size for BOF is 32 with lowest MAPE of 8.5567.

6.4.2.3 CNN

Table 51: Results of SAND Prediction by CNN model in Experiment 15

Test Case	Batch Size	Metrics			
		MSE	MAE	MSE	MAPE
Pre_Close(PC)	8	0.6319	0.5598	0.7949	19.9142
	16	**0.1720**	**0.3147**	**0.4147**	**11.1706**
	32(default)	0.2227	0.3724	0.4719	13.1792
	64	0.2129	0.3405	0.4614	11.9339
	128	0.9701	0.8994	0.9849	31.9958
Primary Technical Indicator(PTI)	8	**1.8631**	**1.0772**	**1.3650**	**38.3035**
	16	2.1265	1.1775	1.4583	41.8988
	32(default)	2.1280	1.1056	1.4588	39.3208
	64	2.0992	1.1761	1.4488	41.8114
	128	1.9174	1.1627	1.3847	41.0185
Advance Technical Indicator(ATI)	8	2.2933	1.4140	1.5144	49.9044
	16	**0.7906**	**0.7112**	**0.8892**	**25.3749**
	32(default)	1.7343	0.9950	1.3169	35.6044
	64	5.4744	2.1384	2.3397	76.2709
	128	12.5260	3.3509	3.5392	119.1695
Best Optimal Features(BOF)	8	2.3291	1.5141	1.5261	53.4964
	16	3.9030	1.9618	1.9756	69.1485
	32(default)	**0.0617**	**0.1974**	**0.2483**	**6.9224**
	64	1.1444	0.9062	1.0698	32.3166
	128	3.5642	1.6749	1.8879	59.5467

Note: Bolded values indicate the best results.
The best results were obtained for 4 test cases for batch size. This information is used in Experiment 16(CNN model) to test different complexity.

From Table 51, it appears that as four features show the different trend of change of the values of MAPE as the batch size increases. The best batch size for Pre_Close is 16 with lowest MAPE of 11.1706. The best batch size for PTI is 8 with lowest MAPE of 38.3035. The best batch size of ATI is 16 with lowest MAPE of 25.3749. The best batch size for BOF is 32 with lowest MAPE of 6.9224.

6.4.2.4 Discussion of Exp 15

Table 52: Results for best parameter in Exp 15

Model	Features	MAPE Exp 14	MAPE Exp 15	Rate Of Change	Parameter
LSTM	Pre_close (PC)	58.8019	51.2340	12.87%	Lr:0.00001, Bs:16
	Primary technical indicator (PTI)	61.2700	44.5605	27.27%	Lr:0.01, Bs:128
	Advance technical indicator (ATI)	69.8294	40.4135	42.13%	Lr:0.01, Bs:128
	Best optimal feature (BOF)	13.3211	**13.1830**	1.04%	Lr:0.001, Bs:128
GRUs	Pre_close (PC)	17.9180	15.3210	14.49%	Lr:0.00001, Bs:64
	Primary technical indicator (PTI)	9.1135	**6.3238**	30.61%	Lr:0.00001, Bs:64
	Advance technical indicator (ATI)	11.7542	8.8777	24.47%	Lr:0.00001, Bs:16
	Best optimal feature (BOF)	8.5567	8.5567	0.00%	Lr:0.001, Bs:32
CNN	Pre_close (PC)	13.1792	11.1706	15.24%	Lr:0.00001, Bs:16
	Primary technical indicator (PTI)	39.3208	38.3035	2.59%	Lr:0.00001, Bs:8
	Advance technical indicator (ATI)	35.6044	25.3749	28.73%	Lr:0.0005, Bs:16
	Best optimal feature (BOF)	6.9224	**6.9224**	0.00%	Lr:0.001, Bs:32

Note: Bolded values indicate the best results.

From Table 52, it appears that all two models (LSTM and CNN) achieved the lowest mean absolute percentage error (MAPE) values when the feature of BOF was used, with 13.3211 and 6.9224 respectively. These models did not undergo a significant rate of change from Experiment 14 to Experiment 15, which suggests that the default batch size (32) was used. However, the LSTM model underwent a slight rate of change when the batch size was hyper-tuned to 128.The GRUs model achieved the lowest mean absolute percentage error (MAPE) values when the feature of PTI was used, instead of BOF with the rate of change of the 30.61 percent from the previous experiments.

When the feature of PC, PTI, and ATI was used in the LSTM model, the MAPE values decreased to 51.2340, 44.5605, 40.4135, with a rate of change of 12.87, 27.27, and 42.13 percent respectively. When the feature of PC, ATI, and BOF was added to the GRUs model, the MAPE values decreased to 15.3210, 8.8777, 8.5567, with a rate of change of 14.49, 24.47, and 0 percent respectively. When the feature of PC, PTI, and ATI used in the CNN model, the MAPE values decreased to 11.1706, 38.3035, 25.3749 with a rate of change of 15.24, 2.59, and 28.73 percent respectively.

6.4.3 Experiment 16 (Complexity)

The results of Experiment 16 have been presented in tabular form, using metrics such as MSE, MAE, RMSE, and MAPE, and have been discussed in later sections. The experiment involved predicting the next 10 days closing price of a metaverse cryptocurrency using LSTM, GRUs, and CNN models with input features including Pre_close(PC), Primary-technical indicator(PTI), Advance technical indicator(ATI), Best Optimal Features(BOF). Hyperparameter tuning was performed by testing different complexities (1, 2, 4) in place of the default value of 3 for each of the models, and the complexity that resulted in the lowest MAPE value was chosen as the setting for future experiments as it resulted in the most accurate predictions according to the MAPE metric, which measures the average absolute percentage error of the model's predictions. The complexities of the LSTM, GRU, and CNN models are presented in the Table 53:

Table 53: Complexity Architecture

Model	Architecture	Complexity
LSTM	First LSTM Layer:32, Second LSTM Layer:16	1
	First LSTM Layer:64, Second LSTM Layer:32	2
	First LSTM Layer:100, Second LSTM Layer:50	3(default)
	First LSTM Layer:128, Second LSTM Layer:64	4
GRUs	First GRUs Layer:32, Second GRUs Layer:16	1
	First GRUs Layer:64, Second GRUs Layer:32	2
	First GRUs Layer:100, Second GRUs Layer:50	3(default)
	First GRUs Layer:128, Second GRUs Layer:64	4
CNN	First Conv1D:8, Second Conv1D:16, Third Conv1D:32, Fourth Dense layer:64	1
	First Conv1D:16, Second Conv1D:32, Third Conv1D:64, Fourth Dense layer:128	2
	First Conv1D:32, Second Conv1D:64, Third Conv1D:128, Fourth Dense layer:256	3(default)
	First Conv1D:64, Second Conv1D:128, Third Conv1D:256, Fourth Dense layer:512	4

6.4.3.1 LSTM

Table 54: Results of SAND Prediction by LSTM model in Experiment 16

Test Case	Complexity	Metrics			
		MSE	MAE	RMSE	MAPE
Pre_Close(PC)	1	1.9132	1.2558	1.3832	44.0841
	2	**0.8531**	**0.8197**	**0.9236**	**29.1245**
	3(default)	2.1596	1.4468	1.4696	51.2340
	4	2.8398	1.6626	1.6852	58.9969
Primary Technical Indicator(PTI)	1	9.1060	2.8721	3.0176	102.0588
	2	2.6942	1.3063	1.6414	46.3500
	3(default)	2.6407	1.2483	1.6250	44.5605
	4	**1.9432**	**1.1294**	**1.3940**	**40.1282**
Advance Technical Indicator(ATI)	1	**0.2344**	**0.4266**	**0.4841**	**15.0261**
	2	2.9042	1.4489	1.7042	51.2229
	3(default)	1.6108	1.1428	1.2692	40.4135
	4	4.9948	2.1720	2.2349	76.9097
Best Optimal Features(BOF)	1	0.5637	0.6813	0.7508	24.2835
	2	1.0840	1.0107	1.0412	35.8393
	3(default)	**0.2710**	**0.3705**	**0.5206**	**13.1830**
	4	0.3268	0.3994	0.5717	14.1095

Note: Bolded values indicate the best results.
The best results were obtained for 4 test cases for complexity. This information is used in Experiment 17(LSTM model) to test different L1 Regularizer.

From Table 54, it appears that as four features show the different trend of change of the values of MAPE as the complexity increases. The best complexity for Pre_Close is 2 with lowest MAPE of 29.1245. The best complexity for PTI is 4 with lowest MAPE of 40.1282. The best complexity of ATI is 1 with lowest MAPE of 15.0261. The best complexity for BOF is 3(default value) with lowest MAPE of 13.1830.

6.4.3.2 GRUs

Table 55: Results of SAND Prediction by GRUs model in Experiment 16

Test Case	Complexity	Metrics			
		MSE	MAE	RMSE	MAPE
Pre_Close(PC)	1	1.0677	0.9212	1.0333	32.4318
	2	**0.1085**	**0.2827**	**0.3293**	**10.0586**
	3(default)	0.2011	0.2011	0.4484	15.3210
	4	0.2322	0.4670	0.4818	16.5985
Primary	1	0.6748	0.7423	0.8215	26.2323
Technical	2	0.1065	0.2843	0.3263	10.0299
Indicator(PTI)	3(default)	**0.0586**	**0.1764**	**0.2421**	**6.3238**
	4	0.1417	0.3357	0.3764	11.8456
Advance	1	4.0745	1.9210	1.9210	67.9510
Technical	2	0.8066	0.8017	0.8981	28.4554
Indicator(ATI)	3(default)	**0.1343**	**0.2470**	**0.3665**	**8.8777**
	4	0.0916	0.2509	0.3027	8.9519
Best	1	10.3370	3.1485	3.2151	111.6965
Optimal	2	7.2683	2.5990	2.6960	92.2919
Features(BOF)	3(default)	**0.0795**	**0.2443**	**0.2820**	**8.5567**
	4	0.2677	0.4082	0.5174	14.5157

Note: Bolded values indicate the best results.
The best results were obtained for 4 test cases for complexity. This information is used in Experiment 17(GRUs model) to test different L1 Regularizer.

From the Table 55, it appears that as four features show the same trend(decrease sharply until a certain point then increase)for MAPE values as the complexity increases. The Pre_Close feature is increase after complexity is 2 ,while the others model is increased after the complexity is 3. The best complexity for Pre_Close is 2 with lowest MAPE of 10.0586. The best complexity for Primary Technical Indicator is 3(default value) with lowest MAPE of 6.3238. The best complexity of Advance Technical Indicator is 3(default value) with lowest MAPE of 8.8777. The best complexity for Best Optimal Feature is 3(default value) with lowest MAPE of 8.5567.

6.4.3.3 CNN

Table 56: Results of SAND Prediction by CNN model in Experiment 16

Test Case	Complexity	Metrics			
		MSE	MAE	RMSE	MAPE
Pre_Close	1	0.8475	0.7346	0.9206	25.7057
	2	1.4553	0.9263	1.2063	32.9677
	3(default)	**0.1720**	**0.3147**	**0.4147**	**11.1706**
	4	0.3417	0.5089	0.5846	17.8830
Primary technical indicator	1	2.1610	1.1281	1.4700	39.1460
	2	2.4999	1.2417	1.5811	43.7578
	3(default)	1.8631	1.0772	1.3650	38.3035
	4	**1.4181**	**0.9295**	**1.1909**	**33.0251**
Advance technical indicator	1	2.3468	1.3133	1.5319	46.1475
	2	**0.4629**	**0.5175**	**0.6803**	**18.0695**
	3(default)	0.7906	0.7112	0.8892	25.3749
	4	7.3511	2.5325	2.7113	89.8734
Best Optimal Features	1	0.2746	0.4193	0.5240	14.5751
	2	1.4208	1.0926	1.1920	38.9207
	3(default)	**0.0617**	**0.1974**	**0.2483**	**6.9224**
	4	0.6112	0.7709	0.7818	27.2449

Note: Bolded values indicate the best results.
The best results were obtained for 4 test cases for complexity. This information is used in Experiment 17(CNN model) to test different L1 Regularizer.

From the Table 56, it appears that as four features show the same trend N shape (increase constantly, decrease sharply and then increase constantly)for MAPE values as the complexity increases. The best complexity for Pre_Close is 3(default) with lowest MAPE of 11.1706. The best complexity for Primary Technical Indicator is 4 with lowest MAPE of 33.0251. The best complexity of Advance Technical Indicator is 2 with lowest MAPE of 18.0695. The best complexity for Best Optimal Feature is 3(default value) with lowest MAPE of 6.9224.

6.4.3.4 Discussion of Exp 16

Table 57: Results for best parameter in Exp 16

Model	Features	MAPE Exp 15	MAPE Exp 16	Rate Of Change	Parameter
LSTM	Pre_close (PC)	51.2340	29.1245	43.15%	Lr:0.00001, Bs:16, Comp:2
	Primary technical indicator (PTI)	44.5605	40.1282	9.95%	Lr:0.01, Bs:128, Comp:4
	Advance technical indicator (ATI)	40.4135	15.0261	62.82%	Lr:0.01, Bs:128, Comp:1
	Best optimal feature (BOF)	13.1830	**13.1830**	0.00%	Lr:0.001, Bs:128, Comp:3
GRUs	Pre_close (PC)	15.3210	10.0586	34.35%	Lr:0.00001, Bs:64, Comp:2
	Primary technical indicator (PTI)	6.3238	**6.3238**	0.00%	Lr:0.00001, Bs:64, Comp:3
	Advance technical indicator (ATI)	8.8777	8.8777	0.00%	Lr:0.00001, Bs:16, Comp:3
	Best optimal feature (BOF)	8.5567	8.5567	0.00%	Lr:0.001, Bs:32, Comp:3
CNN	Pre_close (PC)	11.1706	11.1706	0.00%	Lr:0.00001, Bs:16, Comp:3
	Primary technical indicator (PTI)	38.3035	33.0251	13.78%	Lr:0.00001, Bs:8, Comp:4
	Advance technical indicator (ATI)	25.3749	18.0695	28.79%	Lr:0.0005, Bs:16, Comp:2
	Best optimal feature (BOF)	6.9224	**6.9224**	0.00%	Lr:0.001, Bs:32, Comp:3

Note: Bolded values indicate the best results.

From Table 57, it appears that the LSTM and CNN models both had the lowest MAPE values when using the BOF feature, with values of 13.1830 and 6.9224 respectively. The GRUs model had the lowest MAPE value when using the PTI feature, with a value of 6.3238. These models did not experience significant changes from Experiment 15 to Experiment 16, indicating that the default complexity level was used. When the LSTM model was given the PC, PTI, and ATI features, the MAPE values decreased by 43.15, 9.95, and 62.82 percent respectively. In the GRUs model, the MAPE value decreased by 10.0586 when using the PC feature, but no other changes were observed. In the CNN model, the PTI and ATI features caused the MAPE values to decrease by 13.78 and 28.79 percent respectively, while the PC feature did not have any effect on the model.

6.4.4 Experiment 17 (L1 Regularization)

The results of Experiment 17 have been presented in tabular form, using metrics such as MAPE, and have been discussed in later sections. The experiment involved predicting the next 10 days closing price of a metaverse cryptocurrency using LSTM, GRUs, and CNN models with input features including Pre_close(PC), Primary-technical indicator(PTI), Advance technical indicator(ATI), Best Optimal Features(BOF). Hyperparameter tuning was performed by testing different L1 regularization (0.1, 0.01, 0.005, 0.001, 0.0001) in place of the default value of None(means no L1 Regularizer) multiple layers of these models, and the L1 regularization that resulted in the lowest MAPE value was chosen as the setting for future experiments as it resulted in the most accurate predictions according to the MAPE metric, which measures the average absolute percentage error of the model's predictions. The L1 regularization of the LSTM, GRU, and CNN models are presented in the Table 58:

Table 58: L1 Regularizer Architecture and Layer Set Up

Model	Sequence of layer	Architecture	Remarks				
			1 layer	2 layers	3 layers	4 layers	5 layers
LSTM	1	LSTM	✓	✓	✓	✓	
	2	Dense		✓	✓	✓	
	3	LSTM			✓	✓	
	4	Dropout					
	5	Dense				✓	
GRUs	1	GRUs	✓	✓	✓	✓	
	2	Dense		✓	✓	✓	
	3	GRUs			✓	✓	
	4	Dropout					
	5	Dense				✓	
CNN	1	Conv1D	✓	✓	✓	✓	✓
	2	Conv1D		✓	✓	✓	✓
	3	MaxPooling1D					
	4	Conv1D			✓	✓	✓
	5	Flatten					
	6	Dense				✓	✓
	7	Dropout					
	8	Dense					✓

The LSTM model is composed of five layers, the first of which is an LSTM layer, followed by a dense layer, another LSTM layer, a dropout layer, and an output dense layer. The GRUs model is composed of five layers, the first of which is an GRUs layer, followed by a dense layer, another GRUs layer, a dropout layer, and an output dense layer. In this LSTM and GRUs cases, dropout layer will be excluded from adding the L1 Regularizer. The CNN model is composed of 8 layers, the first of which is an Conv1D layer, followed by another Conv1D layer, MaxPooling1D layer, another Conv1D layer, Flatten layer, Dense layer, Dropout layer and an output dense layer. In this CNN cases, MaxPooling1D, flatten layer and Dropout will be excluded from adding the L1 Regularizer. Adding the L1 Regularizer only to the first layer is referred to as the 1 layer. L1 Regularizer is only added to the second layer is referred to as the 2 layers, and similarly for 3 layers ,4 layers and 5 layers.

6.4.4.1 LSTM

Table 59: Results of SAND Prediction by LSTM model in Experiment 17

Test Case	L1 Regularizer	Metrics (MAPE)			
		LSTM architecture			
		1 layer	2 layers	3 layers	4 layers
Pre_Close(PC)	None(default)	29.1245	29.1245	29.1245	29.1245
	0.1	13.8266	69.0398	68.7962	98.9413
	0.01	63.8877	7.1329	14.4474	14.9541
	0.005	45.3716	45.8422	**4.1331**	5.0997
	0.001	68.3560	74.0014	73.1647	70.0209
	0.0001	94.6492	96.4276	96.1918	98.4351
Primary technical indicator (PTI)	None(default)	40.1282	40.1282	40.1282	40.1282
	0.1	28.8481	64.7807	64.6841	98.5455
	0.01	32.2849	**11.0624**	62.7871	199.4837
	0.005	30.5526	52.3820	91.9244	55.5628
	0.001	31.5806	31.4370	266.5444	160.6556
	0.0001	105.3039	48.4884	49.1280	45.4590
Advance technical indicator (ATI)	None(default)	15.0261	15.0261	15.0261	15.0261
	0.1	80.9712	64.6623	64.6857	98.3941
	0.01	29.3481	63.6246	69.5452	30.3635
	0.005	15.7188	15.3317	26.6305	28.1275
	0.001	26.0431	64.2724	38.9037	**13.6505**
	0.0001	50.3786	53.0167	75.5981	48.0346
Best Optimal Features (BOF)	None(default)	13.1830	13.1830	13.1830	13.1830
	0.1	25.4087	64.9193	64.7638	98.8905
	0.01	21.6764	16.7212	21.9544	6.8040
	0.005	19.5264	15.6033	35.4631	28.8993
	0.001	20.0204	15.1304	**6.7918**	7.1980
	0.0001	32.9481	25.5643	23.6010	22.9718

Note: Bolded values indicate the best results The best results were obtained for 4 test cases for L1 regularization. This information is used in Experiment 18(LSTM model) to test different dropout layer.

From the Table 59, it appears that as four features show the different trend of change of the values of MAPE as the different layers and different L1 regularization of the LSTM model. The best L1 regularization and the layers used are for PC is 0.005 and 3 layers with lowest MAPE of 4.1331 for LSTM. The best L1 regularization and the layers used are for PTI is 0.01and 2 layers with lowest MAPE of 11.0624 for LSTM. The best L1 regularization and the layers used are for ATI is 0.001 and 4 layers with lowest MAPE of 13.6505 for LSTM. The best L1 regularization and the layers used are for BOF is 0.001 and 3 layers with lowest MAPE of 6.7918 for LSTM.

6.4.4.2 GRUs

Table 60: Results of SAND Prediction by GRUs model in Experiment 17

Test Case	L1 Regularizer	Metrics (MAPE)			
		GRUs architecture			
		1 layer	2 layers	3 layers	4 layers
Pre_Close	None(default)	**10.0586**	**10.0586**	**10.0586**	**10.0586**
	0.1	43.2082	66.4311	90.8615	98.6139
	0.01	31.8537	59.5458	75.8357	86.0688
	0.005	19.7030	40.7105	51.2616	59.5521
	0.001	10.8867	11.9698	11.8369	11.6581
	0.0001	10.4821	10.8562	10.8477	10.7633
Primary technical indicator	None(default)	6.3238	6.3238	6.3238	6.3238
	0.1	28.6399	61.5064	90.6691	98.4042
	0.01	13.6307	39.7582	69.4638	82.1831
	0.005	5.6415	10.6283	25.7171	34.5082
	0.001	**4.1629**	4.7507	4.7744	4.9793
	0.0001	5.8892	6.0681	5.8962	5.7832
Advance technical indicator	None(default)	8.8777	8.8777	8.8777	8.8777
	0.1	53.2570	70.7813	91.2438	97.9313
	0.01	40.4599	62.1617	79.1734	86.9159
	0.005	24.8673	43.2127	56.7249	64.2388
	0.001	7.7569	**6.7779**	7.2187	7.8684
	0.0001	9.0296	8.9099	8.7887	8.6982
Best Optimal Features	None(default)	**8.5567**	**8.5567**	**8.5567**	**8.5567**
	0.1	89.4596	165.5212	68.2684	98.9518
	0.01	101.7413	177.2476	60.9309	75.2790
	0.005	110.1714	94.8322	142.8281	126.1277
	0.001	41.0563	72.8049	79.5274	80.3138
	0.0001	12.6531	78.6067	70.8065	57.1684

Note: Bolded values indicate the best results The best results were obtained for 4 test cases for L1 regularization. This information is used in Experiment 18(GRUs model) to test different dropout layer.

From the Table 60, it appears that as four features show the different trend of change of the values of MAPE as the different layers and different L1 regularization of the GRUs model. The best L1 regularization and the layers used are for PC is no L1 regularization with lowest MAPE of 10.0586 for GRUs. The best L1 regularization and the layers used are for PTI is 0.001 and 1 layer with lowest MAPE of 4.1629 for GRUs. The best L1 regularization and the layers used are for ATI is 0.001 and 2 layers with lowest MAPE of 6.7779 for GRUs. The best L1 regularization and the layers used are for BOF is no L1 regularization with lowest MAPE of 8.5567 for GRUs.

6.4.4.3 CNN

Table 61: Results of SAND Prediction by CNN model in Experiment 17

Test Case	L1 Regularizer	Metrics (MAPE)				
		CNN architecture				
		1 layer	2 layers	3 layers	4 layers	5 layers
Pre_Close	None(default)	11.1706	11.1706	11.1706	11.1706	11.1706
	0.1	13.4554	9.6578	13.6592	91.3976	98.9803
	0.01	12.6663	9.6248	22.4999	5.6835	16.2063
	0.005	11.8130	9.9699	23.1783	8.4923	**4.7637**
	0.001	11.1636	10.3378	15.4085	11.3344	10.5099
	0.0001	11.2481	11.0426	11.4352	11.2322	11.2040
Primary technical indicator	None(default)	33.0251	33.0251	33.0251	33.0251	33.0251
	0.1	35.4269	21.1315	41.7847	91.9117	98.6712
	0.01	35.1184	21.1315	20.7786	26.3834	36.5501
	0.005	34.3899	21.9054	20.9808	**17.0175**	20.0996
	0.001	31.9377	26.0806	24.3922	24.2194	24.4812
	0.0001	32.0834	33.2543	26.5986	26.5570	26.5593
Advance technical indicator	None(default)	18.0695	18.0695	18.0695	18.0695	18.0695
	0.1	149.2565	107.8912	49.9739	57.5901	100.3826
	0.01	77.5420	15.8966	18.8626	25.4480	**11.8850**
	0.005	85.8823	20.5343	23.8565	111.0258	75.3381
	0.001	39.0690	38.2239	38.3064	30.6814	31.3210
	0.0001	20.1128	26.8038	19.8652	22.4040	23.1781
Best Optimal Features	None(default)	**6.9224**	**6.9224**	**6.9224**	**6.9224**	**6.9224**
	0.1	25.8897	76.2400	61.4906	68.9502	99.0336
	0.01	32.0930	94.7105	50.2017	78.4151	94.6229
	0.005	15.6612	13.4837	39.0015	50.5812	46.4121
	0.001	13.5333	23.5856	30.0832	24.5241	37.4949
	0.0001	27.4798	7.2158	6.9691	90.0147	11.9132

Note: Bolded values indicate the best results The best results were obtained for 4 test cases for L1 regularization. This information is used in Experiment 18(CNN model) to test different dropout layer.

From the data provided, it appears that as four features show the different trend of change of the values of MAPE as the different layers and different L1 regularization of the CNN model. The best L1 regularization and the layers used are for PC is 0.005 and 5 layers with lowest MAPE of 4.7637. The best L1 regularization and the layers used are for PTI is 0.005 and 4 layers with lowest MAPE of 17.0175. The best L1 regularization and the layers used are for ATI is 0.01 and 5 layers with lowest MAPE of 11.8850. The best L1 regularization and the layers used are for BOF is no L1 regularization with lowest MAPE of 6.9224

6.4.4.4 Discussion of Exp 17

Table 62: Results for best parameter in Exp 17

Model	Features	MAPE Exp 16	MAPE Exp 17	Rate Of Change	Parameter
LSTM	Pre_close (PC)	29.1245	**4.1331**	85.81%	Lr:0.00001, Bs:16, Comp:2, L1:0.005(3 layers)
	Primary technical indicator (PTI)	40.1282	11.0624	72.43%	Lr:0.01, Bs:128, Comp:4, L1:0.01(2 layers)
	Advance technical indicator (ATI)	15.0261	13.6505	9.15%	Lr:0.01, Bs:128, Comp:1, L1:0.001(4 layers)
	Best optimal feature (BOF)	13.183	6.7918	48.48%	Lr:0.001, Bs:128, Comp:3, L1:0.001(3 layers)
GRUs	Pre_close (PC)	10.0586	10.0586	0.00%	Lr:0.00001, Bs:64, Comp:2, L1: None
	Primary technical indicator (PTI)	6.3238	**4.1629**	34.17%	Lr:0.00001, Bs:64, Comp:3, L1:0.001(1 layer)
	Advance technical indicator (ATI)	8.8777	6.7779	23.65%	Lr:0.00001, Bs:16, Comp:3, L1:0.001(2 layers)
	Best optimal feature (BOF)	8.5567	8.5567	0.00%	Lr:0.001, Bs:32, Comp:3, L1:None
CNN	Pre_close (PC)	11.1706	**4.7637**	57.36%	Lr:0.00001, Bs:16, Comp:3, L1:0.005(5 layers)
	Primary technical indicator (PTI)	33.0251	17.0175	48.47%	Lr:0.00001, Bs:8, Comp:4, L1:0.005(4 layers)
	Advance technical indicator (ATI)	18.0695	11.885	34.23%	Lr:0.0005, Bs:16, Comp:2, L1:0.01(5 layers)
	Best optimal feature (BOF)	6.9224	6.9224	0.00%	Lr:0.001, Bs:32, Comp:3, L1:None

Note: Bolded values indicate the best results

From Table 62, it appears that the LSTM and CNN models both had the lowest MAPE values when using the PC feature, with values of 4.1331 and 4.7637 respectively. The GRUs model had the lowest MAPE value when using the PTI feature, with a value of 4.1629. These models experienced significant changes from Experiment 16 to Experiment 17, as the rate of change of 85.81,34.17 and 57.36 percent. When the LSTM model was given the PTI, ATI, and BOF features, the MAPE values decreased by 72.43, 9.15, and 48.48 percent to the value of 11.0624, 13.6505 and 6.7918 respectively. In the GRUs model, the MAPE value decreased to 6.7779 when using the ATI feature, while the others but no other changes were observed. In the CNN model, the PTI and ATI features caused the MAPE values to decrease by 48.47 and 34.23 percent respectively, while the BOF feature did not have any effect on the model.

6.4.5 Experiment 18 (Dropout layer)

The results of Experiment 18 have been presented in tabular form, using metrics such as MSE, MAE, RMSE, and MAPE, and have been discussed in later sections. The experiment involved predicting the next 10 days closing price of a metaverse cryptocurrency using LSTM, GRUs, and CNN models with input features including Pre_close(PC), Primary-technical indicator(PTI), Advance technical indicator(ATI), Best Optimal Features(BOF). Hyperparameter tuning was performed by testing different dropout layer value (0.1,0.3,0.4,0.5) in place of the default value of 0.2 for each of the models, and the dropout layer value that resulted in the lowest MAPE value was chosen as the setting for future experiments as it resulted in the most accurate predictions according to the MAPE metric, which measures the average absolute percentage error of the model's predictions.

6.4.5.1 LSTM

Table 63: Results of SAND Prediction by LSTM model in Experiment 18

Test Case	Drop out layer	Metrics			
		MSE	MAE	RMSE	MAPE
Pre_Close	0.1	0.3854	0.6004	0.6208	21.3374
	0.2(default)	**0.0240**	**0.1194**	**0.1550**	**4.1331**
	0.3	0.1543	0.3664	0.3928	13.0671
	0.4	0.5077	0.6971	0.7125	24.7685
	0.5	0.1536	0.3731	0.3919	13.2836
Primary technical indicator	0.1	1.0440	0.9009	1.0218	31.5845
	0.2(default)	**0.1585**	**0.3181**	**0.3982**	**11.0624**
	0.3	2.5158	1.3178	1.5861	46.4665
	0.4	0.3274	0.4768	0.5722	16.6283
	0.5	1.7253	1.1581	1.3135	40.6312
Advance technical indicator	0.1	0.7635	0.5926	0.8738	21.1067
	0.2(default)	0.2844	0.3858	0.5333	13.6505
	0.3	0.2004	0.4082	0.4477	14.3693
	0.4	**0.0722**	**0.2326**	**0.2687**	**8.1794**
	0.5	5.9081	2.4288	2.4307	85.9423
Best Optimal Features	0.1	0.1307	0.2748	0.3616	9.7632
	0.2(default)	0.0638	0.1916	0.2526	6.7918
	0.3	**0.0489**	**0.1686**	**0.2212**	**5.9280**
	0.4	0.1144	0.3185	0.3382	11.2782
	0.5	0.0685	0.2447	0.2618	8.6898

Note: Bolded values indicate the best results The best results were obtained for 4 test cases for dropout layer. This information is used in Experiment 19(LSTM model) to test different L2 regularization.

From the Table 63, it appears that as four features show the different trend of change of the values of MAPE as the drop out layer value increases of the LSTM model. The best L1 regularization and the layers used are for PC is 0.2 with lowest MAPE of 4.1331 for LSTM. The best L1 regularization and the layers used are for PTI is 0.2 with lowest MAPE of 11.0624 for LSTM. The best L1 regularization and the layers used are for ATI is 0.4 with lowest MAPE of 8.1794 for LSTM. The best L1 regularization and the layers used are for BOF is 0.3 with lowest MAPE of 5.9280 for LSTM.

6.4.5.2 GRUs

Table 64: Results of SAND Prediction by GRUs model in Experiment 18

Test Case	Drop out layer	Metrics			
		MSE	MAE	RMSE	MAPE
Pre_Close	0.1	0.1560	0.3565	0.3950	12.6168
	0.2(default)	0.1085	0.2827	0.3293	10.0586
	0.3	**0.1003**	**0.2689**	**0.3166**	**9.5707**
	0.4	0.1368	0.2695	0.3699	9.5950
	0.5	0.2426	0.3753	0.4925	13.2599
Primary technical indicator	0.1	0.0453	0.1438	0.2128	5.1545
	0.2(default)	**0.0239**	**0.1157**	**0.1545**	**4.1629**
	0.3	0.0348	0.1653	0.1866	5.8677
	0.4	0.0747	0.2505	0.2733	8.8222
	0.5	0.1565	0.3674	0.3956	12.9060
Advance technical indicator	0.1	0.0624	0.2090	0.2497	7.4673
	0.2(default)	**0.0506**	**0.1906**	**0.2250**	**6.7779**
	0.3	0.0703	0.2246	0.2651	7.9350
	0.4	0.1384	0.3218	0.3720	11.3325
	0.5	0.2935	0.4886	0.5418	17.1665
Best Optimal Features	0.1	0.0943	0.2595	0.3070	9.0891
	0.2(default)	**0.0795**	**0.2443**	**0.2820**	**8.5567**
	0.3	0.1175	0.3065	0.3427	10.7313
	0.4	0.0833	0.2726	0.2886	9.5673
	0.5	0.6132	0.7421	0.7831	26.3832

Note: Bolded values indicate the best results The best results were obtained for 4 test cases for dropout layer. This information is used in Experiment 19(GRUs model) to test different L2 regularization.

From the data provided, it appears that as four features show the different trend of change of the values of MAPE as the drop out layer value increases of the GRUs model. The best L1 regularization and the layers used are for PC is 0.3 with lowest MAPE of 9.5707 for GRUs. The best L1 regularization and the layers used are for PTI is 0.2 with lowest MAPE of 4.1629 for GRUs. The best L1 regularization and the layers used are for ATI is 0.2 with lowest MAPE of 6.7779 for GRUs. the best L1 regularization and the layers used are for BOF is 0.2 with lowest MAPE of 8.5567 for GRUs.

6.4.5.3 CNN

Table 65: Results of SAND Prediction by CNN model in Experiment 18

Test Case	Drop out layer	Metrics			
		MSE	MAE	RMSE	MAPE
Pre_Close	0.1	0.0291	0.1387	0.1705	4.8817
	0.2(default)	0.0269	0.1357	0.1641	4.7637
	0.3	0.0255	0.1378	0.1596	4.8333
	0.4	0.0246	0.1361	0.1569	4.7643
	0.5	**0.0234**	**0.1308**	**0.1529**	**4.5688**
Primary technical indicator	0.1	0.3538	0.4897	0.5948	17.2983
	0.2(default)	0.3430	0.4818	0.5857	17.0175
	0.3	0.3497	0.4903	0.5913	17.3126
	0.4	0.3343	0.4782	0.5782	16.8857
	0.5	**0.3250**	**0.4665**	**0.5701**	**16.4728**
Advance technical indicator	0.1	0.3237	0.5005	0.5690	17.9438
	0.2(default)	**0.1535**	**0.3415**	**0.3918**	**11.8850**
	0.3	0.2911	0.4941	0.5395	17.6631
	0.4	4.2788	2.0476	2.0685	72.6313
	0.5	2.4105	1.5291	1.5526	54.2864
Best Optimal Features	0.1	0.2786	0.4472	0.5278	15.9435
	0.2(default)	**0.0617**	**0.1974**	**0.2483**	**6.9224**
	0.3	0.5529	0.6605	0.7435	23.6379
	0.4	0.3956	0.5381	0.6290	19.2873
	0.5	0.5776	0.6223	0.7600	21.7272

Note: Bolded values indicate the best results The best results were obtained for 4 test cases for dropout layer. This information is used in Experiment 19(GRUs model) to test different L2 regularization.

From the Table 65, it appears that as four features show the different trend of change of the values of MAPE as the drop out layer value increases of the CNN model. The best L1 regularization and the layers used are for PC is 0.3 with lowest MAPE of 4.5688. The best L1 regularization and the layers used are for PTI is 0.5 with lowest MAPE of 16.4728. The best L1 regularization and the layers used are for ATI is 0.2 with lowest MAPE of 11.8850. The best L1 regularization and the layers used are for BOF is 0.2 with lowest MAPE of 6.9224.

6.4.5.4 Discussion of Exp 18

Table 66: Results for best parameter in Exp 18

Model	Features	MAPE Exp 17	MAPE Exp 18	Rate Of Change	Parameter
LSTM	Pre_close (PC)	4.1331	**4.1331**	4.09%	Lr:0.00001, Bs:16, Comp:2, L1:0.005(3 layers), D:0.2
	Primary technical indicator (PTI)	11.0624	11.0624	0.00%	Lr:0.01, Bs:128, Comp:4, L1:0.01(2 layers), D:0.2
	Advance technical indicator (ATI)	13.6505	8.1794	40.08%	Lr:0.01, Bs:128, Comp:1, L1:0.001(4 layers), D:0.4
	Best optimal feature (BOF)	6.7918	5.9280	12.72%	Lr:0.001, Bs:128, Comp:3, L1:0.001(3 layers), D:0.3
GRUs	Pre_close (PC)	10.0586	9.5707	4.85%	Lr:0.00001, Bs:64, Comp:2, L1:None, D:0.3
	Primary technical indicator (PTI)	4.1629	**4.1629**	0.00%	Lr:0.00001, Bs:64, Comp:3, L1:0.001(1 layer), D:0.2
	Advance technical indicator (ATI)	6.7779	6.7779	0.00%	Lr:0.00001, Bs:16, Comp:3, L1:0.001(2 layers), D:0.2
	Best optimal feature (BOF)	8.5567	8.5567	0.00%	Lr:0.001, Bs:32, Comp:3, L1:None, D:0.2
CNN	Pre_close (PC)	4.7637	**4.5688**	0.00%	Lr:0.00001, Bs:16, Comp:3, L1:0.005(5 layers), D:0.2
	Primary technical indicator (PTI)	17.0175	16.4728	3.20%	Lr:0.00001, Bs:8, Comp:4, L1:0.005(4 layers), D:0.5
	Advance technical indicator (ATI)	11.885	11.8850	0.00%	Lr:0.0005, Bs:16, Comp:2, L1:0.01(5 layers), D:0.2
	Best optimal feature (BOF)	6.9224	6.9224	0.00%	Lr:0.001, Bs:32, Comp:3, L1:None, D:0.2

From Table 66, LSTM and CNN models both had the lowest MAPE values when using the PC feature, with values of 4.1331 and 4.5688 respectively. The GRUs model had the lowest MAPE value when using the PTI feature, with a value of 4.1629. LSTM did not experience significant changes from Experiment 17 to Experiment 18, indicating that the default drops out layer was used. When the LSTM model was given the ATI, and BOF features, the MAPE values decreased by 40.08 and 12.72 percent to the value of 8.1794 and 5.9280 respectively, while the others but no changes were observed. In the GRUs model, the MAPE value decreased to 9.5707 when using the PC feature, while the others but no changes were observed. In the CNN model, the PTI features caused the MAPE values to decrease by 3.20 percent, while the other features did not have any effect on the model.

6.4.6 Experiment 19 (L2 Regularization)

The results of Experiment 19 have been presented in tabular form, using metrics such as MAPE and have been discussed in later sections. The experiment involved predicting the next 10 days closing price of a metaverse cryptocurrency using LSTM, GRUs, and CNN models with input features including Pre_close(PC), Primary-technical indicator(PTI), Advance technical indicator(ATI), Best Optimal Features(BOF).Hyperparameter tuning was performed by testing different L2 regularization (0.1, 0.01, 0.005, 0.001, 0.0001)in the multiple layers of these models in place of the default value of None(means no L2 Regularizer), and the L2 regularization that resulted in the lowest MAPE value was chosen as the setting for future experiments as it resulted in the most accurate predictions according to the MAPE metric, which measures the average absolute percentage error of the model's predictions. The L2 regularization of the LSTM, GRU, and CNN models are presented in the Table 67 below:

Table 67: L2 Regularization Architecture and Layer Set Up

Model	Sequence of layer	Architecture	L2 Regularizer Remarks				
			1 layer	2 layers	3 layers	4 layers	5 layers
LSTM	1	LSTM	✓	✓	✓	✓	
	2	Dense		✓	✓	✓	
	3	LSTM			✓	✓	
	4	Dropout					
	5	Dense				✓	
GRUs	1	GRUs	✓	✓	✓	✓	
	2	Dense		✓	✓	✓	
	3	GRUs			✓	✓	
	4	Dropout					
	5	Dense				✓	
CNN	1	Conv1D	✓	✓	✓	✓	✓
	2	Conv1D		✓	✓	✓	✓
	3	MaxPooling1D					
	4	Conv1D			✓	✓	✓
	5	Flatten					
	6	Dense				✓	✓
	7	Dropout					
	8	Dense					✓

The LSTM model is composed of five layers, the first of which is an LSTM layer, followed by a dense layer, another LSTM layer, a dropout layer, and an output dense layer. The GRUs model is composed of five layers, the first of which is an GRUs layer, followed by a dense layer, another GRUs layer, a dropout layer, and an output dense layer. In this LSTM and GRUs cases, dropout layer will be excluded from adding the L1 Regularizer. The CNN model is composed of 8 layers, the first of which is an Conv1D layer, followed by another Conv1D layer, MaxPooling1D layer, another Conv1D layer, Flatten layer, Dense layer, Dropout layer and an output dense layer. In this CNN cases, MaxPooling1D, flatten layer and Dropout will be excluded from adding the L1 Regularizer. Adding the L1 Regularizer only to the first layer is referred to as the 1 layer. L1 Regularizer is only added to the second layer is referred to as the 2 layers, and similarly for 3 layers ,4 layers and 5 layers.

6.4.6.1 LSTM

Table 68: Results of SAND Prediction by LSTM model in Experiment 19

Test Case	L2 Regularizer	Metrics (MAPE)			
		LSTM architecture			
		1 layer	2 layers	3 layers	4 layers
Pre_Close	None(default)	**4.1331**	**4.1331**	**4.1331**	**4.1331**
	0.1	24.2225	68.8284	68.8146	68.7971
	0.01	6.8425	22.3447	68.8140	68.7971
	0.005	5.2930	30.2561	68.8119	68.7971
	0.001	4.7561	4.7883	12.4663	68.7971
	0.0001	4.6394	6.6646	9.6457	35.4533
Primary technical indicator	None(default)	**11.0624**	**11.0624**	**11.0624**	**11.0624**
	0.1	37.4774	64.6805	64.6838	64.6775
	0.01	45.0733	64.6877	64.6841	64.6846
	0.005	28.3012	64.6628	64.6838	64.6829
	0.001	26.7170	31.7914	64.6839	64.6844
	0.0001	27.1467	51.3769	86.1477	74.1506
Advance technical indicator	None(default)	**8.1794**	**8.1794**	**8.1794**	**8.1794**
	0.1	61.3594	66.1652	66.1644	66.1646
	0.01	72.4416	69.6160	66.1637	66.1642
	0.005	25.5383	78.2616	87.1311	66.1643
	0.001	32.4992	53.1156	87.9136	73.1209
	0.0001	11.0552	55.2328	71.7604	61.6292
Best Optimal Features	None(default)	6.9224	6.9224	6.9224	6.9224
	0.1	20.3295	30.8161	64.7734	64.7643
	0.01	10.4911	68.7398	67.5982	64.7643
	0.005	7.4333	23.3350	55.2054	59.2440
	0.001	**5.5742**	37.2654	44.9767	47.2825
	0.0001	32.3809	38.4826	49.6548	29.6492

Note: Bolded values indicate the best results The best results were obtained for 4 test cases for L2 regularization layer. This information is used in Experiment 20(LSTM model) to test different epochs.

From the Table 68 above, it appears that as four features show the different trend of change of the values of MAPE as the different layers and different L2 regularization of the LSTM model. The best L2 regularization and the layers used are for PC is no L2 regularization with lowest MAPE of 4.1331 for LSTM. The best L2 regularization and the layers used are for PTI is no L2 regularization with lowest MAPE of 11.0624 for LSTM. The best L2 regularization and the layers used are for ATI is no L2 regularization with lowest MAPE of 8.1794 for LSTM. The best L2 regularization and the layers used are for BOF is 0.001 and 1 layers with lowest MAPE of 5.5742 for LSTM.

6.4.6.2 GRUs

Table 69: Results of SAND Prediction by GRUs model in Experiment 19

Test Case	L2 Regularizer	Metrics (MAPE)			
		GRUs architecture			
		1 layer	2 layers	3 layers	4 layers
Pre_Close	None(default)	**9.5707**	**9.5707**	**9.5707**	**9.5707**
	0.1	13.6848	36.6591	54.6095	65.0474
	0.01	9.9302	15.7350	25.6512	37.4761
	0.005	9.8338	10.9038	15.7856	21.3914
	0.001	9.6872	10.0553	10.5172	10.5389
	0.0001	9.5960	9.6319	9.9020	9.9335
Primary indicator technical	None(default)	**4.1629**	**4.1629**	**4.1629**	**4.1629**
	0.1	4.1551	5.9300	23.1265	52.0644
	0.01	4.5783	3.9053	5.0680	6.0246
	0.005	4.4508	3.9499	4.3420	4.3752
	0.001	4.1574	4.0527	4.1162	4.0402
	0.0001	4.1669	4.1632	4.1613	4.1417
Advance technical indicator	None(default)	6.7779	6.7779	6.7779	6.7779
	0.1	13.3362	44.3193	65.5272	74.5502
	0.01	9.3576	19.1583	34.2084	29.3501
	0.005	8.1546	13.2878	20.4338	29.3501
	0.001	6.9443	8.4087	8.9787	9.3310
	0.0001	6.7931	6.7803	6.6536	**6.6193**
Best Optimal Features	None(default)	**8.5567**	**8.5567**	**8.5567**	**8.5567**
	0.1	75.7800	88.1014	67.0277	68.2599
	0.01	94.4903	72.1888	91.8630	99.5486
	0.005	92.8100	68.8723	79.7719	91.7551
	0.001	98.4840	111.8173	100.5680	79.0441
	0.0001	46.3989	59.4442	95.2032	123.7775

Note: Bolded values indicate the best results The best results were obtained for 4 test cases for L2 regularization layer. This information is used in Experiment 20(GRUs model) to test different epochs.

From the Table 69 above, it appears that as four features show the different trend of change of the values of MAPE as the different layers and different L2 regularization of the GRUs model. The best L2 regularization and the layers used are for PC is no L2 regularization with lowest MAPE of 9.5707 for GRUs. The best L2 regularization and the layers used are for PTI is 0.01 and 2 layers with lowest MAPE of 3.9053 for GRUs. The best L2 regularization and the layers used are for ATI is 0.0001 and 4 layers with lowest MAPE of 6.6193 for GRUs. The best L2 regularization and the layers used are for BOF is no L2 regularization with lowest MAPE of 85567 for GRUs.

6.4.6.3 CNN

Table 70: Results of SAND Prediction by CNN model in Experiment 19

Test Case	L1 Regularizer	Metrics (MAPE)				
		CNN architecture				
		1 layer	2 layers	3 layers	4 layers	5 layers
Pre_Close	None(default)	4.5688	4.5688	4.5688	4.5688	4.5688
	0.1	4.5110	5.6301	6.3713	80.8011	91.6288
	0.01	4.6654	5.2348	4.9440	28.6627	58.4721
	0.005	4.7262	4.9476	4.6077	15.2730	32.6370
	0.001	4.7672	4.7531	4.5855	4.1419	5.0794
	0.0001	4.7561	4.7081	4.7329	4.1600	**4.1206**
Primary technical indicator	None(default)	16.4728	16.4728	16.4728	16.4728	16.4728
	0.1	13.4323	10.2981	9.2544	60.4182	89.1281
	0.01	14.4775	9.9919	9.0010	17.3094	54.7191
	0.005	15.0894	10.3500	9.0762	7.4424	27.5414
	0.001	15.7970	13.5517	10.0529	6.3970	**4.6808**
	0.0001	16.2049	15.6539	15.0120	10.4972	10.0234
Advance technical indicator	None(default)	**11.8850**	**11.8850**	**11.8850**	**11.8850**	**11.8850**
	0.1	72.9707	84.6265	83.8777	83.9217	83.4989
	0.01	60.3319	56.6544	35.9312	84.0961	83.4990
	0.005	55.5799	55.4255	38.0227	27.0953	83.4991
	0.001	37.9157	63.7275	35.7091	44.8197	27.6058
	0.0001	21.0820	31.0224	25.5116	50.1994	44.6244
Best Optimal Features	None(default)	6.9224	6.9224	6.9224	6.9224	6.9224
	0.1	68.3887	102.5366	68.4239	68.9540	68.2335
	0.01	9.7586	64.3504	54.2376	23.9776	31.9667
	0.005	20.3332	40.9284	56.4392	14.2821	15.8167
	0.001	17.1396	14.7101	42.0325	21.3895	21.9205
	0.0001	23.6436	27.1263	28.4605	**4.9980**	17.6058

Note: Bolded values indicate the best results The best results were obtained for 4 test cases for L2 regularization layer. This information is used in Experiment 20(CNN model) to test different epochs.

From the Table 70 above, it appears that as four features show the different trend of change of the values of MAPE as the different layers and different L2 regularization of the CNN model. The best L2 regularization and the layers used are for PC is 0.0001 and 5 layers with lowest MAPE of 4.1206 for CNN. The best L2 regularization and the layers used are for PTI is 0.001 and 5 layers with lowest MAPE of 4.6808 for CNN. The best L2 regularization and the layers used are for ATI is no L2 regularization with lowest MAPE of 11.8850 for CNN. The best L2 regularization and the layers used are for BOF is 0.0001 and 4 layers with lowest MAPE of 4.9980 for CNN.

6.4.6.4 Discussion of Exp 19

Table 71: Results for best parameter in Exp 19

Model	Features	MAPE Exp 18	MAPE Exp 19	Rate Of Change	Parameter
LSTM	Pre_close (PC)	4.1331	**4.1331**	0.00%	Lr:0.00001, Bs:16, Comp:2, L1:0.005(3 layers), D:0.2, L2:None
	Primary technical indicator (PTI)	11.0624	11.0624	0.00%	Lr:0.01, Bs:128, Comp:4, L1:0.01(2 layers), D:0.2, L2:None
	Advance technical indicator (ATI)	8.1794	8.1794	0.00%	Lr:0.01, Bs:128, Comp:1, L1:0.001(4 layers), D:0.4, L2:None
	Best optimal feature (BOF)	5.9280	5.5742	5.97%	Lr:0.001, Bs:128, Comp:3, L1:0.001(3 layers), D:0.3, L2:0.001(1 layer)
GRUs	Pre_close (PC)	9.5707	9.5707	0.00%	Lr:0.00001, Bs:64, Comp:2, L1:None, D:0.3, L2:None
	Primary technical indicator (PTI)	4.1629	**3.9053**	6.19%	Lr:0.00001, Bs:64, Comp:3, L1:0.001(1 layer), D:0.2, L2:0.01(2 layers)
	Advance technical indicator (ATI)	6.7779	6.6193	2.34%	Lr:0.00001, Bs:16, Comp:3, L1:0.001(2 layers), D:0.2, L2:0.0001(4 layers)
	Best optimal feature (BOF)	8.5567	8.5567	0.00%	Lr:0.001, Bs:32, Comp:3, L1:None, D:0.2, L2:None
CNN	Pre_close (PC)	4.7637	**4.1206**	13.50%	Lr:0.00001, Bs:16, Comp:3, L1:0.005(5 layers), D:0.2, L2:0.0001(5 layers)
	Primary technical indicator (PTI)	16.4728	4.6808	71.58%	Lr:0.00001, Bs:8, Comp:4, L1:0.005(4 layers), D:0.5, L2:0.001(5 layers)
	Advance technical indicator (ATI)	11.8850	11.8850	0.00%	Lr:0.0005, Bs:16, Comp:2, L1:0.01(5 layers), D:0.2, L2:None
	Best optimal feature (BOF)	6.9224	4.9980	27.80%	Lr:0.001, Bs:32, Comp:3, L1:None, D:0.2, L2:0.0001(4 layers)

From Table 71 above ,the LSTM and CNN models both had the lowest MAPE values when using the PC feature, with values of 4.1331 and 4.1206 respectively. These models did not experience significant changes from Experiment 18 to Experiment 19, indicating that the default drop out layer value was used. The GRUs model had the lowest MAPE value when using the PTI feature, with a value of 3.9053. When the LSTM model was given the PTI, ATI and BOF features, the MAPE values of PTI and ATI no changes were observed ,while for BOF the MAPE value decreased by 5.97 percent to the value of 5.5742. In the GRUs model, the MAPE value decreased to 9.5707 when using the ATI feature, while the others but no changes were observed. In the CNN model, the PTI and ATI features caused the MAPE values to decrease by 71.58 and 27.80 percent, while the other features did not have any effect on the model.

6.4.7 Experiment 20 (Epochs)

The results of Experiment 20 have been presented in tabular form, using metrics such as MSE, MAE, RMSE, and MAPE, and have been discussed in later sections. The experiment involved predicting the next 10 days closing price of a metaverse cryptocurrency using LSTM, GRUs, and CNN models with input features including Pre_close(PC), Primary-technical indicator(PTI), Advance technical indicator(ATI), Best Optimal Features(BOF). Hyperparameter tuning was performed by testing different epochs (20,40,60,80,100) in place of the default value of 50 for each of the models, and the epochs that resulted in the lowest MAPE value was chosen as the setting for future experiments as it resulted in the most accurate predictions according to the MAPE metric, which measures the average absolute percentage error of the model's predictions.

6.4.7.1 LSTM

Table 72: Results of SAND Prediction by LSTM model in Experiment 20

Test Case	Epochs	Metrics			
		MSE	MAE	RMSE	MAPE
Pre_Close	20	1.1756	1.0757	1.0843	38.1319
	40	0.8801	0.9225	0.9381	32.7406
	50(default)	**0.0240**	**0.1194**	**0.1550**	**4.1331**
	60	0.3299	0.5576	0.5744	19.8238
	80	0.0912	0.2625	0.3020	9.3877
	100	1.8190	1.3316	1.3487	47.2180
Primary technical indicator	20	2.8180	1.6373	1.6787	58.1042
	40	4.3076	1.7300	2.0755	61.1262
	50(default)	**0.1585**	**0.3181**	**0.3982**	**11.0624**
	60	1.1189	0.9349	1.0578	32.8052
	80	0.7289	0.7247	0.8538	25.3327
	100	2.3134	1.4200	1.5210	49.9584
Advance technical indicator	20	**0.0340**	**0.1715**	**0.1845**	**6.0111**
	40	0.0625	0.1884	0.2499	6.6271
	50(default)	0.0722	0.2326	0.2687	8.1794
	60	1.6876	1.2556	1.2991	44.2662
	80	4.7844	2.1675	2.1873	76.5275
	100	6.6275	2.5108	2.5744	88.5439
Best Optimal Features	20	0.4365	0.6530	0.6607	23.1123
	40	0.1615	0.3661	0.4019	12.9447
	50(default)	**0.0429**	**0.1574**	**0.2071**	**5.5742**
	60	0.0513	0.1951	0.2264	6.8738
	80	0.0437	0.1824	0.2090	6.4247
	100	0.0812	0.2509	0.2850	8.8343

Note: Bolded values indicate the best results.

From the Table 72 above, it appears that as four features show the different trend of change of the values of MAPE as the different epochs of the LSTM model. The best epochs used are for PC is 50 with lowest MAPE of 4.1331 for LSTM. The best epochs used are for PTI is 50 with lowest MAPE of 11.0624 for LSTM. The best epochs used are for PTI is 20 with lowest MAPE of 6.0111for LSTM. The best epochs used are for PTI is 50 with lowest MAPE of 5.5742 for LSTM.

6.4.7.2 GRUs

Table 73: Results of SAND Prediction by GRUs model in Experiment 20

Test Case	Epochs	Metrics			
		MSE	MAE	RMSE	MAPE
Pre_Close	20	2.5439	1.5325	1.5950	54.2169
	40	0.2865	0.4069	0.5352	14.4401
	50(default)	0.1003	0.2689	0.3166	9.5707
	60	0.0547	0.1965	0.2338	7.0185
	80	0.0363	0.1428	0.1906	5.0941
	100	**0.0289**	**0.1317**	**0.1701**	**4.7208**
Primary technical indicator	20	1.1320	1.0057	1.0640	35.4466
	40	0.0486	0.1561	0.2205	5.5359
	50(default)	0.0234	0.1092	0.1528	3.9053
	60	0.0168	0.0928	0.1296	3.3440
	80	**0.0162**	**0.0879**	**0.1273**	**3.1813**
	100	0.0245	0.1218	0.1567	4.3733
Advance technical indicator	20	2.8967	1.6806	1.7020	59.3072
	40	0.2067	0.3797	0.4546	13.3440
	50(default)	**0.0511**	**0.1864**	**0.2261**	**6.6193**
	60	0.0599	0.1949	0.2447	6.9894
	80	0.1820	0.3662	0.4267	13.0371
	100	0.3458	0.5443	0.5881	19.3058
Best Optimal Features	20	4.2846	2.0042	2.0699	71.0994
	40	1.1771	0.9762	1.0849	34.7630
	50(default)	0.0795	0.2443	0.2820	8.5567
	60	**0.0654**	**0.1901**	**0.2557**	**6.7104**
	80	0.1113	0.2963	0.3336	10.5142
	100	0.1736	0.3807	0.4166	13.5375

From Table 73 above, it appears that as four features show the different trend of change of the values of MAPE as the different epochs of the GRUs model. The best epochs used are for PC is 100 with lowest MAPE of 4.7208 for GRUs. The best epochs used are for PTI is 80 with lowest MAPE of 3.1813 for GRUs. The best epochs used are for ATI is 50 with lowest MAPE of 6.7779 for GRUs. The best epochs used are for BOF is 60 with lowest MAPE of 6.7104 for GRUs.

6.4.7.3 CNN

Table 74: Results of SAND Prediction by CNN model in Experiment 20

Test Case	Epochs	Metrics			
		MSE	MAE	RMSE	MAPE
Pre_Close	20	0.0972	0.2645	0.3117	9.3393
	40	0.0209	0.1305	0.1444	4.5728
	50(default)	0.0204	0.1178	0.1428	4.1206
	60	0.0231	0.1259	0.1520	4.4343
	80	0.0181	0.1185	0.1347	4.1428
	100	**0.0194**	**0.1118**	**0.1393**	**3.8879**
Primary technical indicator	20	0.1869	0.3561	0.4324	12.6124
	40	0.0493	0.1623	0.2221	5.7707
	50(default)	0.0366	0.1317	0.1914	4.6808
	60	0.0192	0.0940	0.1387	3.3215
	80	0.0116	0.0806	0.1076	2.8260
	100	**0.0117**	**0.0810**	**0.1084**	**2.8091**
Advance technical indicator	20	2.6207	1.5948	1.6188	56.1878
	40	0.2661	0.4616	0.5159	16.1089
	50(default)	0.1535	0.3415	0.3918	11.8850
	60	0.7972	0.8697	0.8929	30.9283
	80	7.6812	2.7588	2.7715	97.7319
	100	**0.0794**	**0.2120**	**0.2818**	**7.5929**
Best Optimal Features	20	0.0991	0.2788	0.3148	9.7209
	40	0.3148	1.6257	1.6283	57.3752
	50(default)	**0.0239**	**0.1403**	**0.1547**	**4.9980**
	60	0.6911	0.8258	0.8313	29.1339
	80	0.2368	0.4766	0.4866	16.7991
	100	0.2043	0.4149	0.4520	14.6679

From the Table 74 above, it appears that as four features show the different trend of change of the values of MAPE as the different epochs of the CNN model. The best epochs used are for PC is 100 with lowest MAPE of 3.8879 for CNN. The best epochs used are for PTI is 80 with lowest MAPE of 2.8091 for CNN. The best epochs used are for ATI is 50 with lowest MAPE of 7.5929 for CNN. The best epochs used are for BOF is 60 with lowest MAPE of 4.9980 for CNN.

6.4.7.4 Discussion of Exp 20

Table 75: Results for best parameter in Exp 20

Model	Features	MAPE Exp.19	MAPE Exp 20	Rate Of Change	Parameter
LSTM	Pre_close (PC)	4.1331	**4.1331**	0.00%	Lr:0.00001Bs:16, Comp:2, L1:0.005(3 layers), D:0.2, L2:None, Epoch:50
	Primary technical indicator (PTI)	11.0624	11.0624	0.00%	Lr:0.01, Bs:128, Comp:4, L1:0.01(2 layers), D:0.2, L2:None, Epoch:50
	Advance technical indicator (ATI)	8.1794	6.0111	26.51%	Lr:0.01, Bs:128, Comp:1, L1:0.001(4 layers), D:0.4, L2:None, Epoch:20
	Best optimal feature (BOF)	5.5742	5.5742	0.00%	Lr:0.001, Bs:128, Comp:3, L1:0.001(3 layers), D:0.3, L2:0.001(1 layer), Epoch:50
GRUs	Pre_close (PC)	9.5707	4.7208	50.67%	Lr:0.00001, Bs:64, Comp:2, L1:None, D:0.3, L2:None, Epoch:100
	Primary technical indicator (PTI)	3.9053	**3.1813**	18.54%	Lr:0.00001, Bs:64, Comp:3, L1:0.001(1 layer), D:0.2, L2:0.01(2 layers), Epoch:80
	Advance technical indicator (ATI)	6.6193	6.6193	0.00%	Lr:0.00001, Bs:16, Comp:3, L1:0.001(2 layers), D:0.2, L2:0.0001(4 layers), Epoch:50
	Best optimal feature (BOF)	8.5567	6.7104	21.58%	Lr:0.001, Bs:32, Comp:3, L1:None, D:0.2, L2:None, Epoch:60
CNN	Pre_close (PC)	4.1206	**3.8879**	5.65%	Lr:0.00001, Bs:16, Comp:3, L1:0.005(5 layers), D:0.2, L2:0.0001(5 layers), Epoch:100
	Primary technical indicator (PTI)	4.6808	2.8091	39.99%	Lr:0.00001, Bs:8, Comp:4, L1:0.005(4 layers), D:0.5, L2:0.001(5 layers), Epoch:100
	Advance technical indicator (ATI)	11.8850	7.5929	36.11%	Lr:0.0005, Bs:16, Comp:2, L1:0.01(5 layers), D:0.2, L2:None, Epoch:100
	Best optimal feature (BOF)	4.9980	4.9980	0.00%	Lr:0.001, Bs:32, Comp:3, L1:None, D:0.2, L2:0.0001(4 layers), Epoch:50

The LSTM and CNN models both had the lowest MAPE values when using the PC feature, with values of 4.1331 and 3.8879 respectively. LSTM model did not experience significant changes from Experiment 19 to Experiment 20, indicating that the default epochs was used. The GRUs model had the lowest MAPE value when using the PTI feature, with a value of 3.1813. When the LSTM model was given the PTI,ATI and BOF features, the MAPE values of PTI and BOF no changes were observed ,while for ATI the MAPE value decreased by 26.51 percent to the value of 6.0111. In the GRUs model, the MAPE value decreased to 9.5707 when using the ATI feature, while the others but no changes were observed. In the CNN model, the PTI and ATI features caused the MAPE values to decrease by 71.58 and 27.80 percent, while the other features did not have any effect on the model.

6.5 Results for Hyperparameter Tuning Experiments (14-20)

Table 76: Overall Result for experiments 1 to 13

Currencies	Model	Experiments	Metrics			
			MSE	MAE	MSE	MAPE
Sand	LSTM	PC	**0.0240**	**0.1194**	**0.1550**	**4.1331**
		PTI	0.1585	0.3181	0.3982	11.0624
		ATI	0.0340	0.1715	0.1845	6.0111
		BOF	0.0429	0.1574	0.2071	5.5742
	Grus	PC	0.0289	0.1317	0.1701	4.7208
		PTI	**0.0162**	**0.0879**	**0.1273**	**3.1813**
		ATI	0.0511	0.1864	0.2261	6.6193
		BOF	0.0654	0.1901	0.2557	6.7104
	CNN	PC	0.0194	0.1118	0.1393	3.8879
		PTI	**0.0117**	**0.0810**	**0.1084**	**2.8091**
		ATI	0.0794	0.2120	0.2818	7.5929
		BOF	0.0239	0.1403	0.1547	4.9980
SLP	LSTM	PC	0.0025	0.0504	0.0504	286.8540
		PTI	0.0019	0.0435	0.0436	246.7296
		ATI	0.0045	0.0674	0.0674	382.6539
		BOF	**0.0011**	**0.0335**	**0.0336**	**190.4078**
	Grus	PC	0.0041	0.0642	0.0643	364.9714
		PTI	**0.0034**	**0.0586**	**0.0587**	**333.2947**
		ATI	0.0100	0.0994	0.0998	564.2115
		BOF	0.0147	0.1210	0.1214	686.5053
	CNN	PC	0.0041	0.0636	0.0637	361.1419
		PTI	**0.0008**	**0.0275**	**0.0276**	**156.6376**
		ATI	0.0017	0.0405	0.0408	231.1213
		BOF	0.0071	0.0809	0.0842	458.1446
MANA	LSTM	PC	4.2919	2.0701	2.0717	94.7860
		PTI	3.9088	1.9754	1.9771	90.4437
		ATI	**0.0643**	**0.2400**	**0.2535**	**10.8895**
		BOF	3.9347	1.9819	1.9836	90.7435
	Grus	PC	0.2466	0.4872	0.4966	22.2541
		PTI	1.8298	1.3430	1.3527	61.4701
		ATI	2.9615	1.7172	1.7209	78.6571
		BOF	**0.0873**	**0.2766**	**0.2954**	**12.7124**
	CNN	PC	1.4416	1.1951	1.2007	54.6514
		PTI	4.2372	2.0568	2.0585	94.1800
		ATI	1.9462	1.3747	1.3951	62.7056
		BOF	**0.8682**	**0.9273**	**0.9317**	**42.4119**

Note: Bolded values indicate the best results.

Table 76 above summarizes overall result for experiments 14 to 20. Experiments using LSTM, GRU, and CNN models to predict the closing price of SAND cryptocurrency using PC, PTI, ATI, and BOF features have recorded MAPE values of 4.1331, 11.0624, 6.0111 and 5.5742(LSTM), 4.7208, 3.1813, 6.6193 and 6.7104 (GRUs) and 3.8879, 2.8091, 7.5929 and 4.9980(CNN).

After the hyperparameter tuning, PC, PTI, ATI, and BOF features have achieved very good accuracy and record very low MAPE values means the hyperparameter tuning is very effective to make the models generalize well to the unseen data and avoid the overfitting. Within the total 12 test cases, the CNN with PTI feature have the lowest value of MAPE (2.8091) and then follow by the GRUs with PTI feature(3.1813) and LSTM with PC feature(4.1331). This suggests that the PTI feature have the strongest prediction power to the closing price of SAND cryptocurrency when using the hyperparameter stated in the experiments. The 3 models using BOF features have the good MAPE value but is no low as the PTI and PC features stated above. This suggests that the PTI and PC feature, when fine-tuned with specific techniques, contain more relevant information for the task of predicting the closing price of SAND than BOF. This also indicates the hyperparameter tuning stated in the Experiments 14 to 20 have no so effective to the BOF features compared with the PTI and PC features.

Experiments using LSTM, GRU, and CNN models to predict the closing price of SLP as the test case to test the compatibility of the SAND models after hyperparameter tuning to another cryptocurrencies model. By using PC, PTI, ATI, and BOF features, the results have recorded MAPE values of 286.8540, 246.7296, 382.6539 and 190.4078(LSTM), 364.9714, 333.2947, 564.2115 and 686.5053 (GRUs) and 361.1419, 156.6376, 231.1213 and 458.1446 (CNN). This indicating that the models have a low accuracy in predicting the closing price of SLP.

Experiments using LSTM, GRU, and CNN models to predict the closing price of MANA as another test case to test the compatibility of the SAND models to another cryptocurrencies model. By using PC, PTI, ATI, and BOF features, the results have recorded MAPE values of 94.7860, 90.4437, 10.8895 and 90.7435 (LSTM), 22.2541, 61.4701, 78.6571 and 12.7124 (GRUs) and 54.6514, 94.1800, 62.7056 and 42.4119 (CNN). The results have recorded a range of MAPE values, with some being relatively low and others being relatively high. This means that the models that were trained on the SAND cryptocurrency may not be as compatible with the MANA cryptocurrency, and that further adjustments or retraining may be necessary to improve the prediction accuracy for MANA. The features used in these experiments is not as informative for the MANA cryptocurrency as they were for SAND. There are several reasons why a model that performs well on predicting the closing price of SLP and MANA not performed as well on SAND:

- Different Market Dynamics: The market dynamics for SLP and MANA with SAND is different, which could affect the performance of the model. For example, the volatility, trading volume SLP and MANA is different with the SAND causing the distribution of the data is differ with SAND make it harder to predict SLP and MANA price.

- Feature Relevance: The features used to train the SAND models may be more relevant for SAND compared to the MANA and SLP. The feature selection method needed to customise for the MANA and SLP to prevent the overfitting to the dataset.

- Model Complexity: The model architecture and its complexity in SAND is suite perfectly for the dataset of MANA and SLP. The different set of architecture and complexity is required to predict the price of MANA and SLP.

6.6 Conclusion

Figure 72: Results for best parameter in All Experiments

Cryptocurrencies	Model	Feature	Parameter Used	MAPE
SAND	LSTM	PC	Lr:0.00001, Bs:16, Comp:2, L1:0.005(3 layers), D:0.2, L2:None, Epoch:50	4.1331
	GRUs	PTI	Lr:0.00001, Bs:64, Comp:3, L1:0.001(1 layer), D:0.2, L2:0.01(2 layers), Epoch:80	3.1813
	CNN	PC	Lr:0.00001, Bs:16, Comp:3, L1:0.005(5 layers), D:0.2, L2:0.0001(5 layers), Epoch:100	**2.8091**
SLP	LSTM	BOF	Lr:0.001, Bs:128, Comp:3, L1:0.001(3 layers), D:0.3, L2:0.001(1 layer), Epoch:50	190.4078
	GRUs	PTI	Lr:0.00001, Bs:64, Comp:3, L1:0.001(1 layer), D:0.2, L2:0.01(2 layers), Epoch:80	333.2947
	CNN	PTI	Lr:0.00001, Bs:8, Comp:4, L1:0.005(4 layers), D:0.5, L2:0.001(5 layers), Epoch:100	**156.6376**
MANA	LSTM	ATI	Lr:0.01, Bs:128, Comp:1, L1:0.001(4 layers), D:0.4, L2:None, Epoch:20	**10.8895**
	GRUs	BOF	Lr:0.001, Bs:32, Comp:3, L1:None, D:0.2, L2:None, Epoch:60	12.7124
	CNN	BOF	Lr:0.001, Bs:32, Comp:3, L1:None, D:0.2, L2:0.0001(4 layers), Epoch:50	42.4119

Notes: Bolded values indicate the best results.

The SAND cryptocurrency achieved lowest MAPE value as 2.8091 by using CNN as model ,Previous Close(PC) as features ,learning rate of 0.00001, batch size of 16, complexity architecture of 3, L1 Regularization of 0.05 in 5 layers of architecture ,dropout layer value of 0.2, L2 Regularization of 0.0001 in 5 layers of architecture and number of epochs used is 50. The SLP cryptocurrency achieved lowest MAPE value as

156.6376 by using CNN as model ,Primary Technical Indicator(PTI) as features ,learning rate of 0.00001, batch size of 8, complexity architecture of 4, L1 Regularization of 0.005 in 4 layers of architecture, dropout layer value of 0.5, L2 Regularization of 0.001 in 5 layers of architecture and number of epochs used is 100. The MANA cryptocurrency achieved lowest MAPE value as 10.8895 by using LSTM as model ,Advance Technical Indicator(ATI) as features ,learning rate of 0.001, batch size of 128, complexity architecture of 3, L1 Regularization of 0.001 in 4 layers of architecture, dropout layer value of 0.4, no L2 Regularization is used, and number of epochs used is 50.

CHAPTER 7

CONCLUSION

7.1 Introduction

The chapter provides an overview of the project's purpose, objectives and achievements, main findings, limitations and potential for future work. This chapter is divided into 5 sections. An overview of the chapter is provided in the first section. Section 2 details the project summary of FYP 2. Section 3 details the project objectives and achievements of FYP 2. Section 4 describes main finding of this project. Section 4 the future work that and limitation of this project.

7.2 Project Summary

In this project, the use of various machine learning techniques, such as LSTM, GRUs, and CNN, is demonstrated for forecasting the next 10 days of metaverse cryptocurrency prices. The existing dataset has been curated and pre-processed to enhance and increase the number of dependent features. The experiments 1-13 is the feature selection experiments that find out the four dependent features which are "Previous Closing," "Primary Technical Indicator," "Advance Technical Indicator," and "Best Optimal Features” that will be use as the input of following experiments. The parameters used for these three models were standardized to include 50 epochs, a batch size of 32, a Mean Square Error as the loss function, and the Adam as optimizer. Following feature selection, different hyperparameters, including learning rate, batch size, complexity, L1 Regularization, dropout layer value, L2 Regularization, and epochs, were tested in experiments 14-20 to determine the optimal parameters for the LSTM, GRU, and CNN models. These experiments utilized the SAND dataset and were tested using SLP and MANA.

7.3 Project Objectives and Achievements

The objective 1 of the project, which was to curate and modify the existing metaverse-based cryptocurrencies' prices datasets and examine the performance and efficiency of using different features of datasets to forecast metaverse-based cryptocurrencies' prices, was achieved through the completion of 20 experiments. The first 13 experiments were conducted to select various features of the datasets and evaluate their efficiency in predicting prices. Four features were selected for the three models used to predict metaverse-based cryptocurrencies prices, which were "Previous Closing," "Primary Technical Indicator," "Advance Technical Indicator," and "Best Optimal Features". The "Best Optimal Features" bThe last six experiments of the 20 were conducted to improve the performance of the three models (LSTM, GRUs, and CNN) in predicting the closing prices of SAND. The performance and efficiency of the SAND models in predicting SLP and MANA prices were also evaluated to achieve this objective.

The objective 2 of the project, which was to design and implement the Convolutional neural networks (CNN), Long short-term memory (LSTM), and Gated recurrent units (GRUs) machine learning algorithms in the predictive models to forecast metaverse-based cryptocurrencies closing prices, was achieved through the completion of 20 experiments. The CNN, LSTM, and GRUs algorithms were used in every experiment, and hyperparameter tuning was done based on the three models.

The objective 3 of the project, which was to evaluate the performance and efficiency of the machine learning models by using Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE), was achieved through the completion of 20 experiments. The evaluation metrics used for all experiments were MAE, MSE, RMSE and MAPE.

The MAPE value of 2.8091 and 3.1813 was achieved by using the Primary Technical Indicator in the CNN and GRUs models, respectively, followed by 4.1331 using the LSTM model. A total of 12 test case models based on the previous closing price, primary technical indicator, advanced technical indicator, and best optimal features were developed for the SAND cryptocurrency prediction, and they all achieved less than 11.0624 of MAPE error. The models developed for the SAND cryptocurrency prediction were also tested with the SLP and MANA cryptocurrencies to achieve the objectives 1, 2, and 3.

7.4 Project Main Finding

In this section, the project main finding will be discussed with details.T The Best Optimal Feature (BOF) feature consistently results in the lowest mean absolute percentage error (MAPE) value when using the LSTM, GRU, and CNN models when predicting the SAND before in the hyperparameter tuning as the value as the MAPE values recorded as 13.3211, 8.5567 and 6.9224 after the experiments 13. The BOF But after the experiments 20, the MAPE value is recorded as 5.5742, 6.7104, 4.9980, the low rate of change of percentage means the hyperparameter tuning in the experiments does not have significant effects.

The best model to predict SAND is using CNN as model, Previous Close(PC) as features, learning rate of 0.00001, batch size of 16, complexity architecture of 3, L1 Regularization of 0.05 in 5 layers of architecture ,dropout layer value of 0.2, L2 Regularization of 0.0001 in 5 layers of architecture and number of epochs used is 50 with the MAPE of 2.8091.The best model to predict SLP is using CNN as model CNN as model ,Primary Technical Indicator(PTI) as features ,learning rate of 0.00001, batch size of 8, complexity architecture of 4, L1 Regularization of 0.005 in 4 layers of architecture, dropout layer value of 0.5, L2 Regularization of 0.001 in 5 layers of architecture and number of epochs used is 100 with the MAPE of 156.6376.The best model to predict MANA is LSTM as model ,Advance Technical Indicator(ATI) as features ,learning rate of 0.001, batch size of 128, complexity architecture of 3, L1

Regularization of 0.001 in 4 layers of architecture, dropout layer value of 0.4, no L2 Regularization is used, and number of epochs used is 50 with the MAPE of 10.8895.

7.5 Limitation and Future Work

In this section, some limitations and future works of the project were discussed. The LSTM, GRUs, and CNN models used in the 12 test cases achieved very low MAPE values in the prediction of SAND, but moderate MAPE values for MANA and high MAPE values for SLP. This was caused by different market dynamics, feature relevance, and data distribution and quantity. Additionally, SLP is a very low-value metaverse-based cryptocurrency, and MAE, MSE, RMSE, and MAPE can be unreliable when working with low-volume data or small values, as it can lead to large percentage errors. Furthermore, the LSTM and GRUs models are a type of recurrent neural network that processes input sequences in a particular order, and the same problem happened in CNN as CNN recognizes patterns in data, the patterns that are recognized can depend on the order of the input elements. The model's output will change when the order of the input matrix is changed. Therefore, it is important to ensure that the input matrix is always followed with a particular order in future work.

In the future, the focus of this project can expand widely and deeply. First, researchers can conduct more experiments regarding combination of various pre-processing to improve the MAE, MSE, RMSE, and MAPE of the models. To enhance the performance of the model on the SLP and MANA datasets, researchers can consider using hyperparameter tuning specifically tailored for these datasets. This could help to fill the gap identified in the limitations discussed above by using dataset-specific pre-processing techniques, adapting the model architecture and modification of LSTM, GRU and CNN models and using dataset-specific evaluation metrics and hyperparameter tuning technique for for datasets with low-volume data or small values. Instead of using the hyperparameter tuning experiments stated above, researchers can consider using other hyperparameter tuning techniques in future work. It is also suggested that in future works, the researcher can use real-life data for back testing. This is because as the cryptocurrency market is constantly evolving, the use of real-life data can provide a more accurate representation of the market dynamics and enable the researcher to find the optimal solution. Additionally, back testing with real-life data can also help to improve the generalization of the model to unseen data, making it more adaptable to changing market conditions

REFERENCES

Abdulmonem, M. H., EssamEddeen, J., Zakhari, M. H., Hanafi, S., & Mostafa, H. (2020, December). Hardware Acceleration of Dash Mining Using Dynamic Partial Reconfiguration on the ZYNQ Board. In 2020 32nd International Conference on Microelectronics (ICM) (pp. 1-4). IEEE.

Akyildirim, E., Cepni, O., Corbet, S., & Uddin, G. S. (2021). Forecasting mid-price movement of Bitcoin futures using machine learning. Annals of Operations Research, 1-32.

Akyildirim, E., Goncu, A., & Sensoy, A. (2021). Prediction of cryptocurrency returns using machine learning. Annals of Operations Research, 297(1), 3-36.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET) (pp. 1-6). Ieee.

Alharby, M., & Van Moorsel, A. (2017). Blockchain-based smart contracts: A systematic mapping study. arXiv preprint arXiv:1710.06372.

Arnott, R., Harvey, C. R., & Markowitz, H. (2019). A backtesting protocol in the era of machine learning. The Journal of Financial Data Science, 1(1), 64-74.

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.

Bignell, D. E. & Eggleton, P. 1998. Termites. In P. Calow, (ed.) Encyclopedia of Ecology and Environmental Management. Oxford. Blackwell Scientific, pp. 744-746

Borges, T. A., & Neves, R. F. (2020). Ensemble of machine learning algorithms for cryptocurrency investment with different data resampling methods. Applied Soft Computing, 90, 106187.

Brocke, J. V., Simons, A., Niehaves, B., Niehaves, B., Reimer, K., Plattfaut, R., & Cleven, A. (2009). Reconstructing the giant: On the importance of rigour in documenting the literature search process.

Chen, S., & He, H. (2018, October). Stock prediction using convolutional neural network. In IOP Conference series: materials science and engineering (Vol. 435, No. 1, p. 012026). IOP Publishing.

Chen, Z., Lam, O., Jacobson, A., & Milford, M. (2014). Convolutional neural network-based place recognition. arXiv preprint arXiv:1411.1509.

Choo, K. K. R. (2015). Cryptocurrency and virtual currency: Corruption and money laundering/terrorism financing risks? In Handbook of digital currency (pp. 283-307). Academic Press.

Chowdhury, R., Rahman, M. A., Rahman, M. S., & Mahdy, M. R. C. (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.

Chowdhury, R., Rahman, M. A., Rahman, M. S., & Mahdy, M. R. C. (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.

Crosby, M., Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). Blockchain technology: Beyond bitcoin. Applied Innovation, 2(6-10), 71.

Crosby, M., Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). Blockchain technology: Beyond bitcoin. Applied Innovation, 2(6-10), 71.

Derbentsev, V., Datsenko, N., Babenko, V., Pushko, O., & Pursky, O. (2020, October). Forecasting Cryptocurrency Prices Using Ensembles-Based Machine Learning Approach. In 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T) (pp. 707-712). IEEE.

Derbentsev, V., Datsenko, N., Babenko, V., Pushko, O., & Pursky, O. (2020, October). Forecasting Cryptocurrency Prices Using Ensembles-Based Machine Learning Approach. In 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T) (pp. 707-712). IEEE

Felizardo, L., Oliveira, R., Del-Moral-Hernandez, E., & Cozman, F. (2019, October). Comparative study of bitcoin price prediction using WaveNets, recurrent neural networks and other machine learning methods. In 2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC) (pp. 1-6). IEEE.

Felizardo, L., Oliveira, R., Del-Moral-Hernandez, E., & Cozman, F. (2019, October). Comparative study of bitcoin price prediction using WaveNets, recurrent neural networks and other machine learning methods. In 2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC) (pp. 1-6). IEEE

Freeda, S. E., Selvan, T. E., & Hemanandhini, I. G. (2021, December). Prediction of Bitcoin Price using Deep Learning Model. In 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 1702-1706). IEEE.

Gidea, M., Goldsmith, D., Katz, Y., Roldan, P., & Shmalo, Y. (2020). Topological recognition of critical transitions in time series of cryptocurrencies. Physica A: Statistical mechanics and its applications, 548, 123843.

Gu, S., Kelly, B. T., & Xiu, D. (2019). Empirical asset pricing via machine learning. Chicago Booth Research Paper, (18-04), 2018-09.'

Hitam, N. A., Ismail, A. R., & Saeed, F. (2019). An optimized support vector machine (SVM) based on particle swarm optimization (PSO) for cryptocurrency forecasting. Procedia Computer Science, 163, 427-433.

Hitam, N. A., Ismail, A. R., & Saeed, F. (2019). An optimized support vector machine (SVM) based on particle swarm optimization (PSO) for cryptocurrency forecasting. Procedia Computer Science, 163, 427-433.

Indulkar, Y. (2021, March). Time Series Analysis of Cryptocurrencies Using Deep Learning & Fbprophet. In 2021 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 306-311). IEEE.

Indulkar, Y. (2021, March). Time Series Analysis of Cryptocurrencies Using Deep Learning & Fbprophet. In 2021 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 306-311). IEEE.

Jakubowicz, E., & Abdelfattah, E. (2021, October). The Rise and Fall of Bitcoin: Predicting Market Direction Using Machine Learning Models. In 2021 IEEE

12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 0491-0495). IEEE.

Jaquart, P., Dann, D., & Martin, C. (2020). Machine Learning for Bitcoin Pricing-A Structured Literature Review. In Wirtschaftsinformatik (Zentrale Tracks) (pp. 174-188).

Jeon, H. J., Youn, H. C., Ko, S. M., & Kim, T. H. (2022). Blockchain and AI Meet in the Metaverse. Advanceds in the Convergence of Blockchain and Artificial Intelligence, 73.

Kappos, George, Haaroon Yousaf, Mary Maller, and Sarah Meiklejohn. "An empirical analysis of anonymity in zcash." In 27th USENIX Security Symposium (USENIX Security 18), pp. 463-477. 2018

Karakoyun, E. S., & Cibikdiken, A. O. (2018, May). Comparison of arima time series model and lstm deep learning algorithm for bitcoin price forecasting. In The 13th multidisciplinary academic conference in Prague (Vol. 2018, pp. 171 180).

Kim, H. M., Bock, G. W., & Lee, G. (2021). Predicting Ethereum prices with machine learning based on Blockchain information. Expert Systems with Applications, 184, 115480.

Kraaijeveld, O., & De Smedt, J. (2020). The predictive power of public Twitter sentiment for forecasting cryptocurrency prices. Journal of International Financial Markets, Institutions and Money, 65, 101188.

Kraft, D. (2016). Difficulty control for blockchain-based consensus systems. Peer-to-peer Networking and Applications, 9(2), 397-413.

Kypriotaki, K., Zamani, E., & Giaglis, G. (2015, April). From bitcoin to decentralized autonomous corporations-extending the application scope of decentralized peer-to-peer networks and blockchains. In International conference on enterprise information systems (Vol. 2, pp. 284-290). SciTePress.

Lee, L.-H., Braud, T., Zhou, P., Wang, L., Xu, D., Lin, Z., Kumar, A., Bermejo, C., and Hui, P. (2021). All one needs to know about metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda.arXiv preprint arXiv:2110.05352

Li, J., & Zhang, D. (2020). Multi-pose Face Recognition Based on Convolutional Neural Network. Journal of Computers, 31(1), 225-231.

Lobban, T. (2021). BITCOIN as an alternative investment. TAXtalk, 2021(88), 8-9.

Lu, P., Song, B., & Xu, L. (2021). Human face recognition based on convolutional neural network and augmented dataset. Systems Science & Control Engineering, 9(sup2), 29-37.

Madakam, S., & Kollu, S. (2020). Blockchain Technologies Fundamentals Perceptions, Principles, Procedures and Practices. Prajnan, 48(4).

Mohapatra, S., Ahmed, N., & Alencar, P. (2019, December). Kryptooracle: A real time cryptocurrency price prediction platform using twitter sentiments.In 2019 IEEE International Conference on Big Data (Big Data) (pp. 5544-5551).IEEE

Mystakidis, S. (2022). Metaverse. Encyclopedia, 2(1), 486-497.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review, 21260.

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

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.

Peng, Y., Albuquerque, P. H. M., de Sá, J. M. C., Padula, A. J. A., & Montenegro, M. R. (2018). The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with Support Vector Regression. Expert Systems with Applications, 97, 177-192

Petrovic, A., Strumberger, I., Bezdan, T., Jassim, H. S., & Nassor, S. S. (2021,November). Cryptocurrency Price Prediction by Using Hybrid Machine Learning and Beetle Antennae Search Approach.In 2021 29th Telecommunications Forum (TELFOR) (pp. 1-4). IEEE.

Petrovic, A., Strumberger, I., Bezdan, T., Jassim, H. S., & Nassor, S. S. (2021,November). Cryptocurrency Price Prediction by Using Hybrid Machine Learning and Beetle Antennae Search Approach. In 2021 29th Telecommunications Forum (TELFOR) (pp. 1-4). IEEE.

Phaladisailoed, T., & Numnonda, T. (2018, July). Machine learning models comparison for bitcoin price prediction. In 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE) (pp. 506-511). IEEE.

Pilkington, M. (2016). Blockchain technology: principles and applications. In Research handbook on digital transformations. Edward Elgar Publishing.

Politis, A., Doka, K., & Koziris, N. (2021, May). Ether price prediction using advancedd deep learning models. In 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 1-3). IEEE.

Rane, P. V., & Dhage, S. N. (2019, March). Systematic erudition of bitcoin price prediction using machine learning techniques. In 2019 5th International Conference on Advancedd Computing & Communication Systems (ICACCS) pp. 594-598). IEEE.

Rathan, K., Sai, S. V., & Manikanta, T. S. (2019, April). Crypto-currency price prediction using decision tree and regression techniques. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 190-194). IEEE.

Samaddar, M., Roy, R., De, S., & Karmakar, R. (2021, February). A Comparative Study of Different Machine Learning Algorithms on Bitcoin Value Prediction. In 2021 International Conference on Advanceds in Electrical, Computing, Communication and Sustainable Technologies (ICAECT) (pp. 1-7). IEEE.

Samaddar, M., Roy, R., De, S., & Karmakar, R. (2021, February). A Comparative Study of Different Machine Learning Algorithms on Bitcoin Value Prediction. In 2021 International Conference on Advanceds in Electrical, Computing, Communication and Sustainable Technologies (ICAECT) (pp. 1-7). IEEE.

Shahbazi, Z., & Byun, Y. C. (2021). Improving the Cryptocurrency Price Prediction Performance Based on Reinforcement Learning. IEEE Access, 9, 162651-162659.

Shi, Z., Shi, M., & Li, C. (2017, May). The prediction of character based on recurrent neural network language model. In 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS) (pp. 613-616). IEEE

Sharma, N., Jain, V., & Mishra, A. (2018). An analysis of convolutional neural networks for image classification. Procedia computer science, 132, 377-384.

Sun, X., Liu, M., & Sima, Z. (2020). A novel cryptocurrency price trend forecasting model based on LightGBM. Finance Research Letters, 32, 101084

Tiwari, R. G., Agarwal, A. K., Kaushal, R. K., & Kumar, N. (2021, October). Prophetic Analysis of Bitcoin price using Machine Learning Approaches. In 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC) (pp. 428-432). IEEE.

Waheeb, W., Shah, H., Jabreel, M., & Puig, D. (2020, October). Bitcoin Price Forecasting: A Comparative Study Between Statistical and Machine Learning Methods. In 2020 2nd International Conference on Computer and Information Sciences (ICCIS) (pp. 1-5). IEEE.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. MIS quarterly, xiii-xxiii.

Wüst, K., & Gervais, A. (2018, June). Do you need a blockchain?. In 2018 Crypto Valley Conference on Blockchain Technology (CVCBT) (pp. 45-54). IEEE.

Yang, T. H., Tseng, T. H., & Chen, C. P. (2016, November). Recurrent neural network-based language models with variation in net topology, language, and granularity. In 2016 International Conference on Asian Language Processing (IALP) (pp. 71-74). IEEE.

Yiying, W., & Yeze, Z. (2019, March). Cryptocurrency price analysis with artificial intelligence. In 2019 5th International Conference on Information Management (ICIM) (pp. 97-101). IEEE.

Yogeshwaran, S., Kaur, M. J., & Maheshwari, P. (2019, April). Project based learning: predicting bitcoin prices using deep learning. In 2019 IEEE Global Engineering Education Conference (EDUCON)(pp. 1449-1454). IEEE.

Zhang, H., Qu, Z., Yuan, L., & Li, G. (2017, March). A face recognition method based on LBP feature for CNN. In 2017 IEEE 2nd Advancedd Information Technology, Electronic and Automation Control Conference (IAEAC) (pp. 544-547). IEEE.

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, 183, 115378.

Zhu, X., & Bain, M. (2017). B-CNN: branch convolutional neural network for hierarchical classification. arXiv preprint arXiv:1709.09890.

APPENDICES

Appendices A: Proposal Revision/Progress Revision

REVIEWER 1/EXAMINER 1

Comments	Reply to Comments
1. It is an interesting research Idea, and I believe it can be finished within time.	1. Thank you for the comments.

REVIEWER 2/EXAMINER 2

Comments	Reply to Comments
1. It is an interesting research Idea, and I believe it can be finished within time.	1. Thank you for the comments.

REVIEWER 3/EXAMINER 3

Comments	Reply to Comments
1. Excellent proposal. The project seems very interesting. Overall, the proposal is Ok. However, the literature can be further supported by more recent authentic work in the field. The methodology can also be improved. Some novel deep learning algorithms and latest APIs (i.e., ensemble and hybrid learning models) can help achieve great results.	1. The more literature review works was included. 2. For methodology, the LSTM, CNN and GRUs was used. The hybrid-model such as CNN-LSTM was not used as the method and maybe can used as the future work since it is out of original project scope.

REVIEWER 4/EXAMINER 4

Comments	**Reply to Comments**
1. The first project objective can combine with the second project object. Needed to find another new objective. 2. The project scope needed to redefine. 3. The problem statements need to improve to show the metaverse cryptocurrencies is no same as the cryptocurrencies.	1. The first objective was found and stated in the report. 2. The project scope was redefined in the report. 3. The citation was included inside the report about the different of volatility of the metaverse cryptocurrencies and traditional cryptocurrencies.

Comments	**Reply to Comments**
1. Why no use PCA analysis for the feature selection? 2. The presentation can make in clearer way as no using the loop to present the experiments 4-13. 3. Why not do the k-fold validation?	1. The principal component analysis is reducing the dimensions of the dataset. PCA is a linear technique, and it may not be the best approach if the relationship between the features and the target variable is non-linear that might be happen in the dataset. 2. The correction will do with the presentation slide to make sure the presentation is clearer. 3. In time series data, the observations are not independent and identically distributed, as the observations are correlated with one another and the order in which they were collected matters.

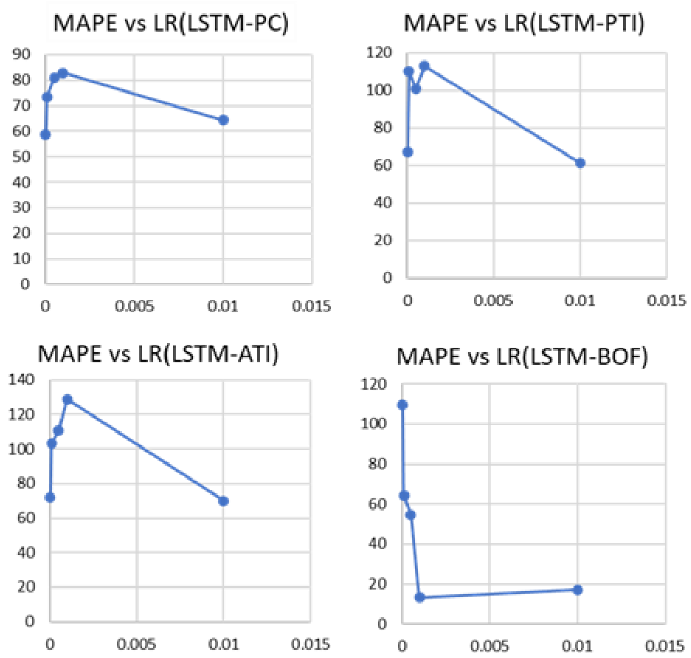
Appendices B: Meeting Log Screenshot

	Date		Status
1 OPEN	**07-Jun-2022** 03:42PM	**Discussed:** discuss the slide presentation and report write up about the full experiment **Next Action:** Finalize the report write up and slide presentation **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED
2 OPEN	**02-Jun-2022** 10:58AM	**Discussed:** Present the results for the third full experiment and discuss the method third full experiment **Next Action:** Planning and prepare the slide presentation and report write up about the full experiment **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED
3 OPEN	**27-May-2022** 03:54PM	**Discussed:** Discussed: Present the results for the second full experiment and discuss the method third full experiment **Next Action:** Planning and prepare the third full experiment **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED
4 OPEN	**20-May-2022** 02:21AM	**Discussed:** Present the results for the first full experiment and discuss the method of second full experiment **Next Action:** Planning and prepare the second full experiment **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED
5 OPEN	**17-May-2022** 01:53AM	**Discussed:** Discuss the following topic 1. Corrections after interim 2. Preparation for first full experiment 3. Planning for remainder of experiments	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED
6 OPEN	**21-Apr-2022** 08:37PM	**Discussed:** Discuss and present about the presentation slide that are used in Week 7 project 1 presentation **Next Action:** Do the improvement and adjustment based on the suggestion of supervisor **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED
7 OPEN	**14-Apr-2022** 07:31PM	**Discussed:** Discuss and present about the progression at the part of writing literature review and system implementation . **Next Action:** Do the adjustment about the interim report based on the supervisor comment **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED
8 OPEN	**07-Apr-2022** 06:49PM	**Discussed:** Discuss and present about the progression at the part of writing literature review and system implementation . **Next Action:** Complete the literature review and progress to the methodology. **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED
9 OPEN	**31-Mar-2022** 11:34AM	**Discussed:** Discuss and present progression on the FYP1 literature review, systems implementation, and any relevant corrections required for new comments just received **Next Action:** Progress the literature review and do the adjustment based on the advise of superviser **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED

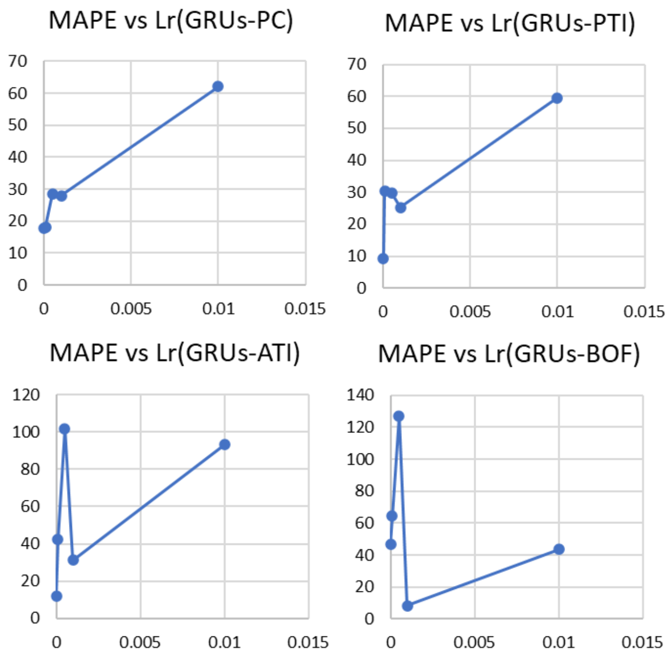
	Date		Status
1 OPEN	**21-Jan-2023** 12:04PM	**Discussed:** Discuss progression of the report writing. Discuss the comment and suggestion given by the examiner and panel. **Next Action:** Finalize the report writing Do the correction ad improvement of the report writing given by the examiner and panel **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED
2 OPEN	**13-Dec-2022** 02:38PM	**Discussed:** Discuss about the following experiment and the progression of the report writing. Discuss the comment and suggestion given by the examiner and panel **Next Action:** Complete the experiments and the report writing Do the correction ad importment of the report writing given by the examiner and panel **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED
3 OPEN	**21-Nov-2022** 03:47PM	**Discussed:** Discuss about the following experiment and the progression of the report writing. **Next Action:** Complete the experiments and the report writing **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED
4 OPEN	**02-Nov-2022** 09:17PM	**Discussed:** Discuss about the following experiment and the progression of the mini review. Ask for the suggestion from the supervisor about the problem of the overfitting of the model and the ways to modify it **Next Action:** Design the following experiment to select to best optimum feature selection and using the different training/validation test set split for the model training. **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED
5 OPEN	**30-Oct-2022** 10:31AM	**Discussed:** Discuss about the following experiment and the progression of the mini review **Next Action:** Design the following experiment to select to best optimum feature selection for the model trainning. **Lecturer Comment:**	**Supervisor:** JASON TEO TZE WI ACCEPTED **Student:** ACCEPTED

Appendices C: MAPE Plots for Experiments (14-20)

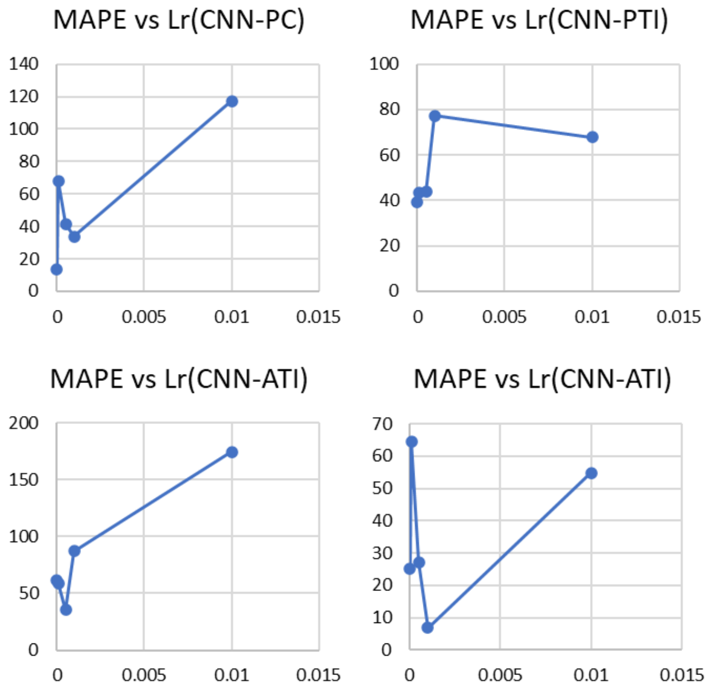
Experiment 14(Learning Rate)



Appendices Figure 1: MAPE vs Lr(LSTM) for Experiment 14

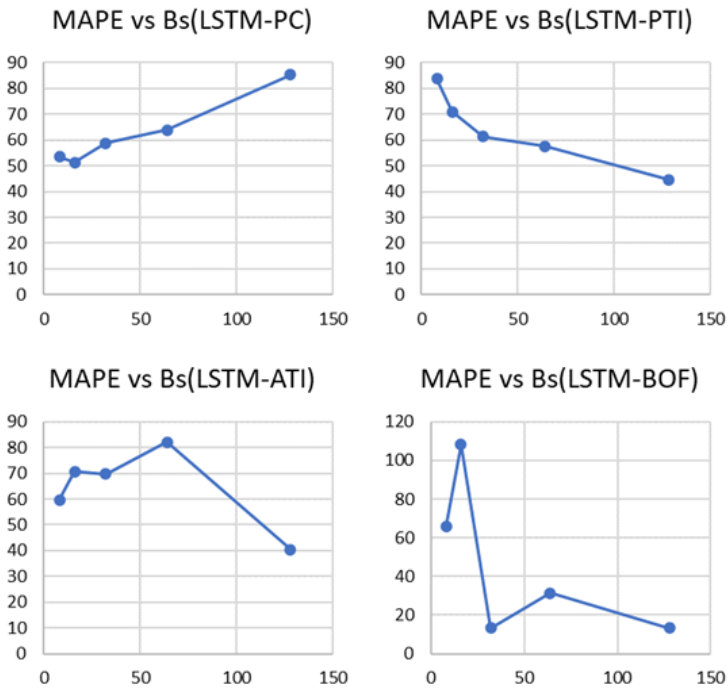


Appendices Figure 2: MAPE vs Lr(LSTM) for Experiment 14

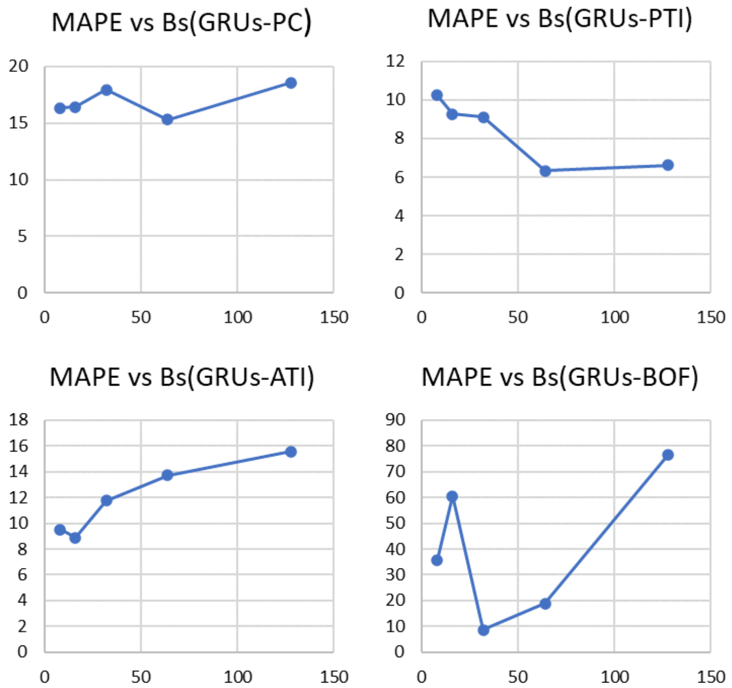


Appendices Figure 3:MAPE vs Lr(CNN) for Experiment 14

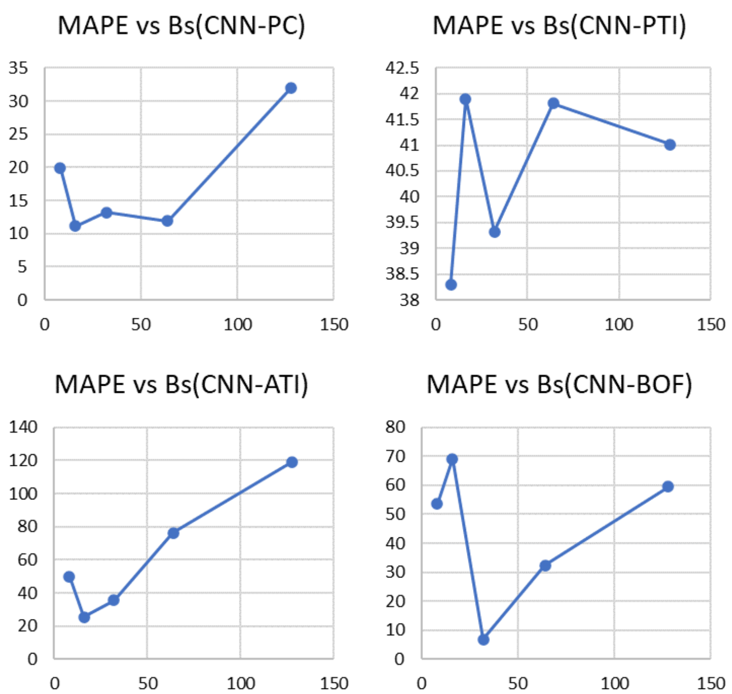
Experiment 15(Batch Size)



Appendices Figure 4: MAPE vs Bs(LSTM) for Experiment 15

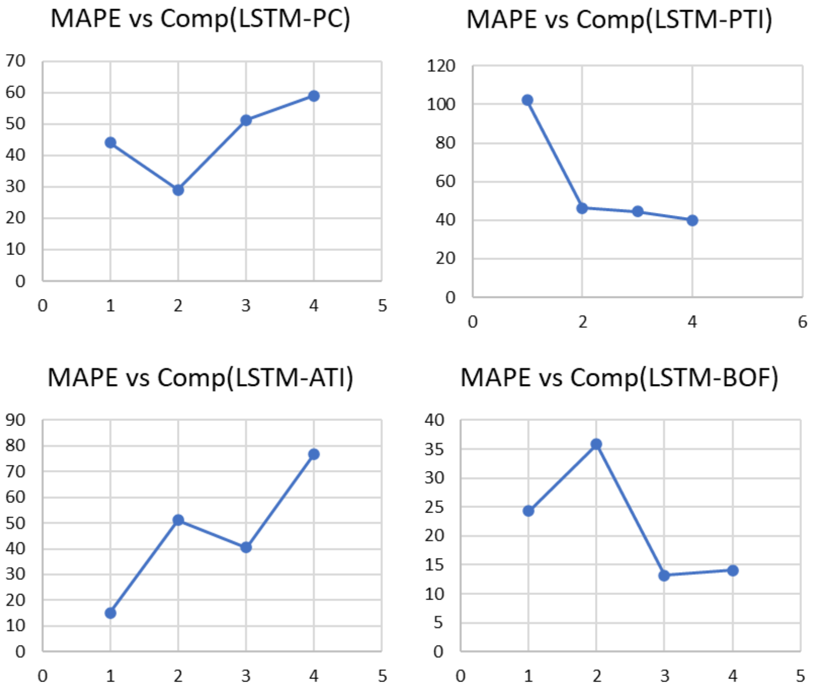


Appendices Figure 5:MAPE vs Bs(GRUs) for Experiment 15

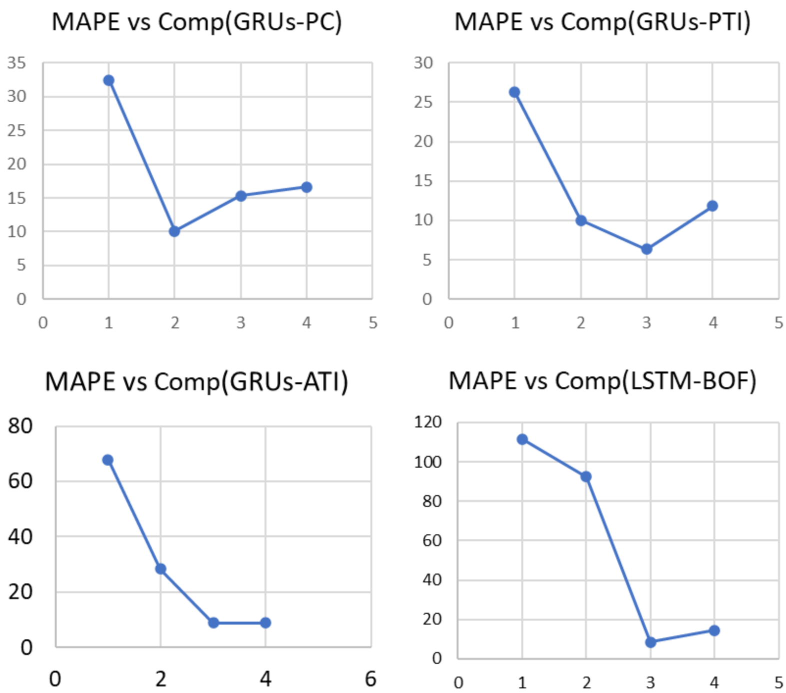


Appendices Figure 6: MAPE vs Bs(CNN) for Experiment 15

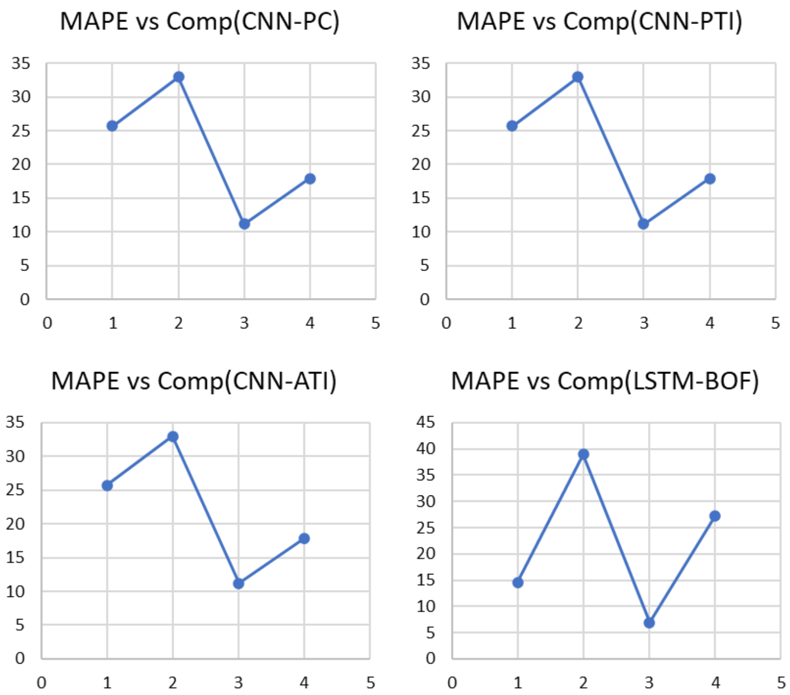
Experiment 16(Complexity)



Appendices Figure 7: MAPE vs Comp(LSTM) for Experiment 16

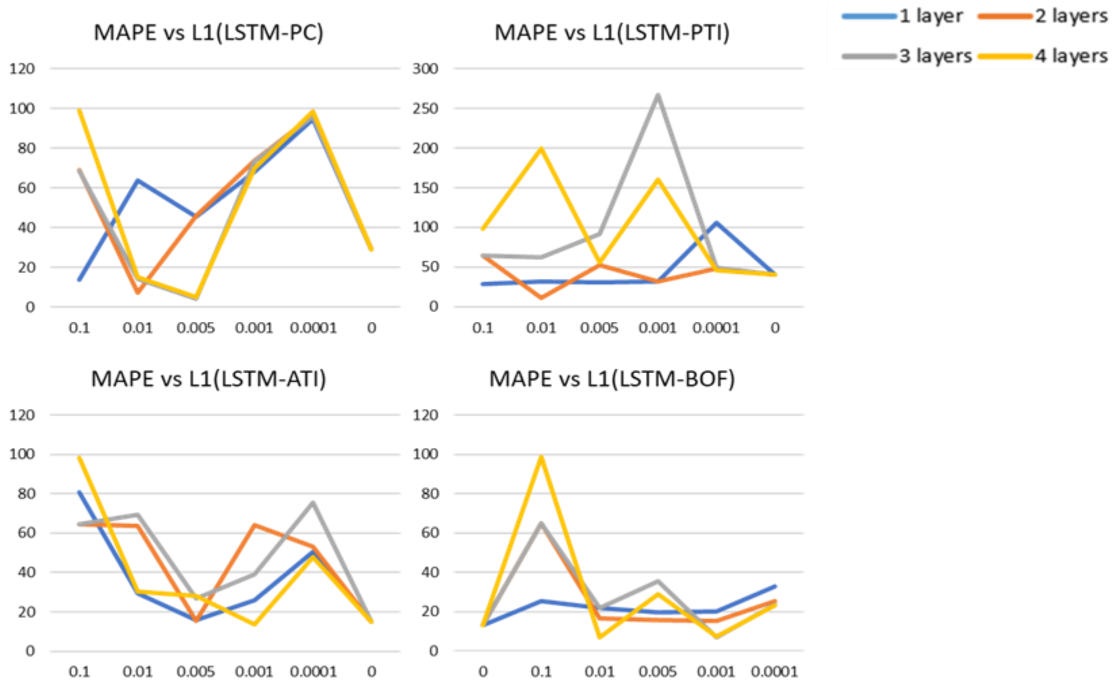


Appendices Figure 8: MAPE vs Comp(GRUs) for Experiment 16

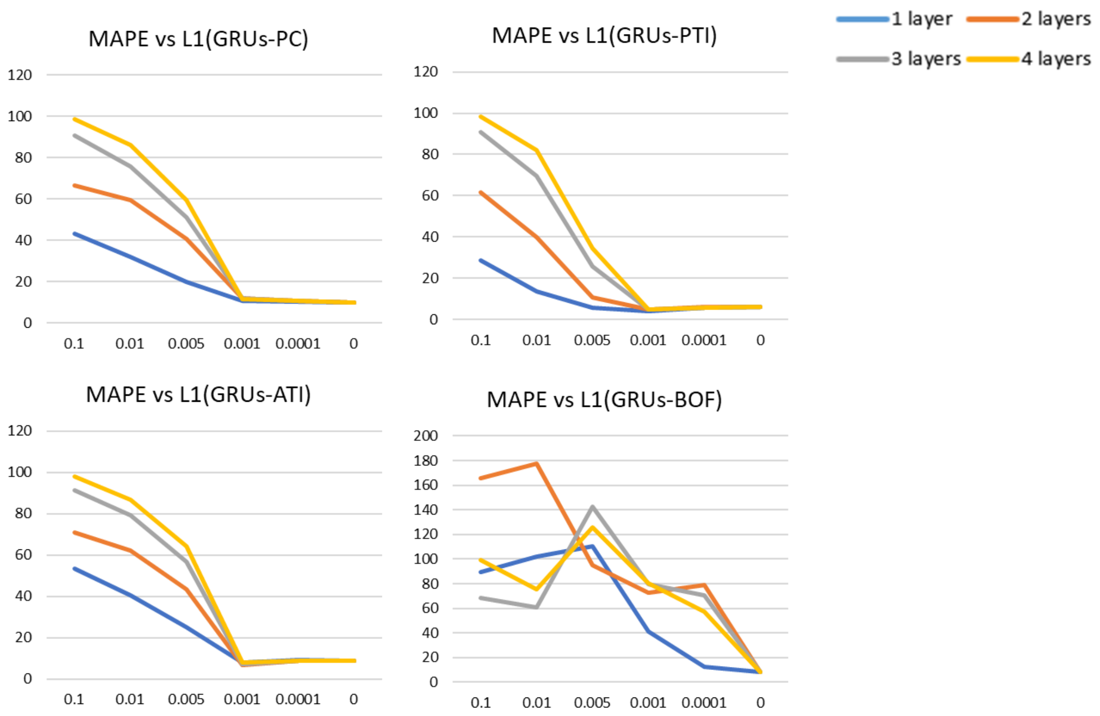


Appendices Figure 9: MAPE vs Comp(CNN) for Experiment 16

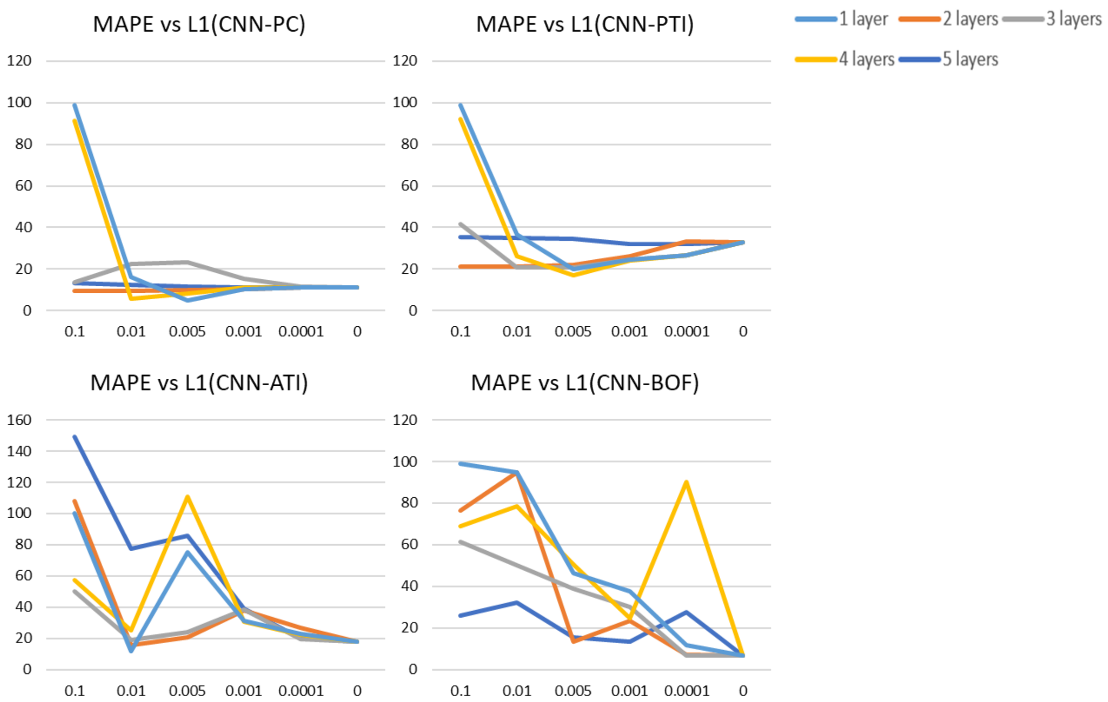
Experiment 17(L1 Regularization)



Appendices Figure 10: MAPE vs L1(LSTM) for Experiment 17

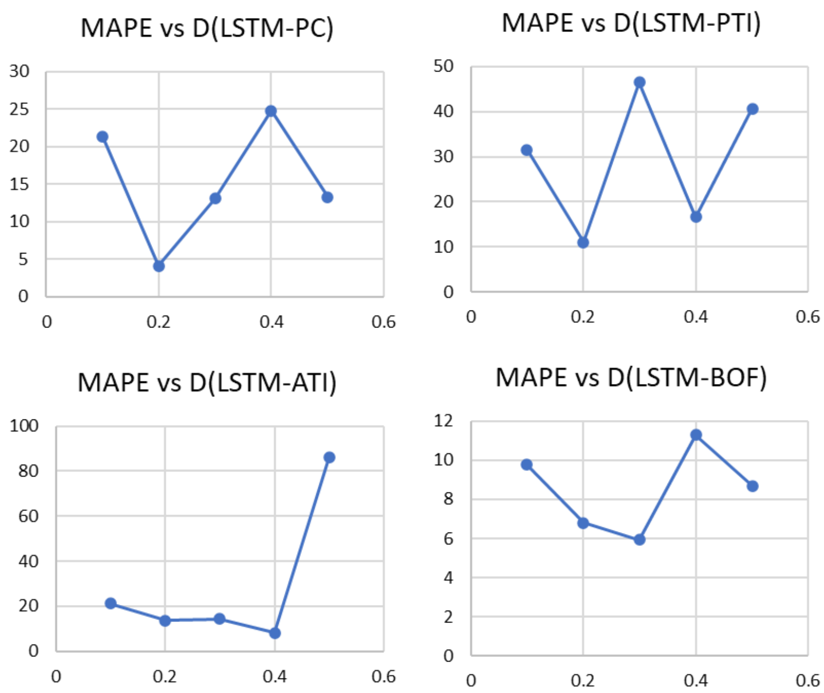


Appendices Figure 11: MAPE vs L1(GRUs) for Experiment 17

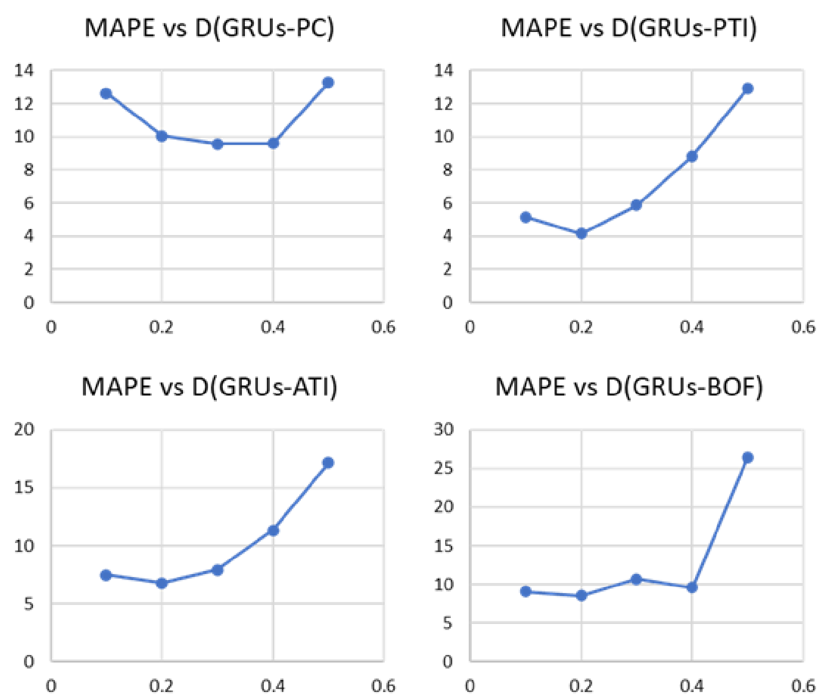


Appendices Figure 12:MAPE vs L1(CNN) for Experiment 17

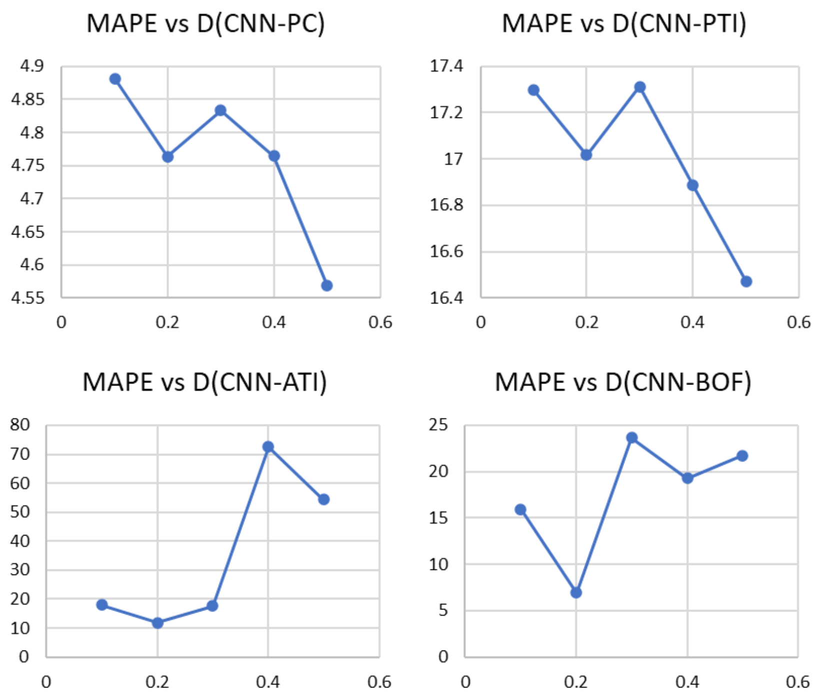
Experiment 18(Dropout layer)



Appendices Figure 13: MAPE vs D(LSTM) for Experiment 18

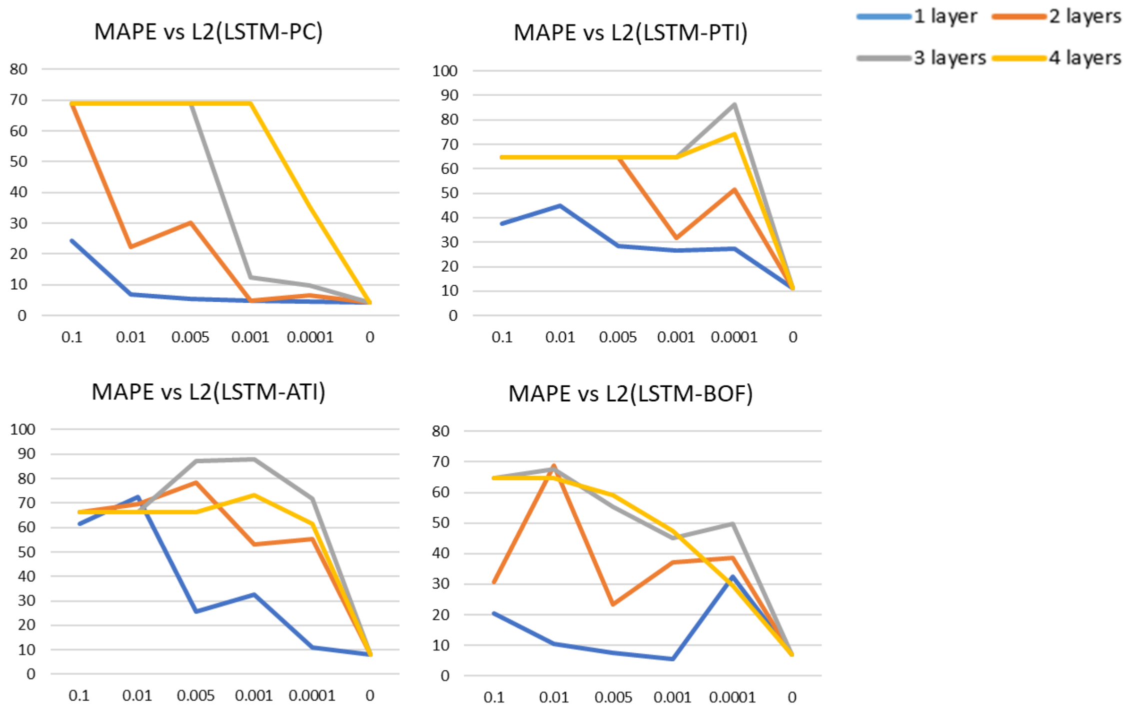


Appendices Figure 14: MAPE vs D(GRUs) for Experiment 18

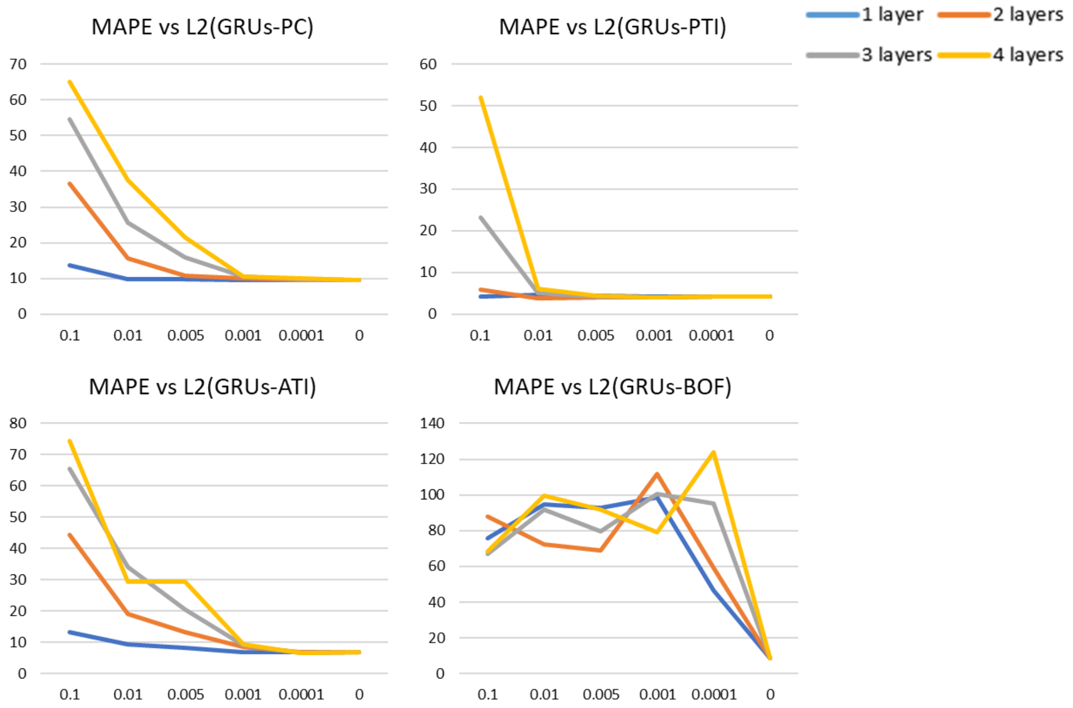


Appendices Figure 15: MAPE vs D(CNN) for Experiment 18

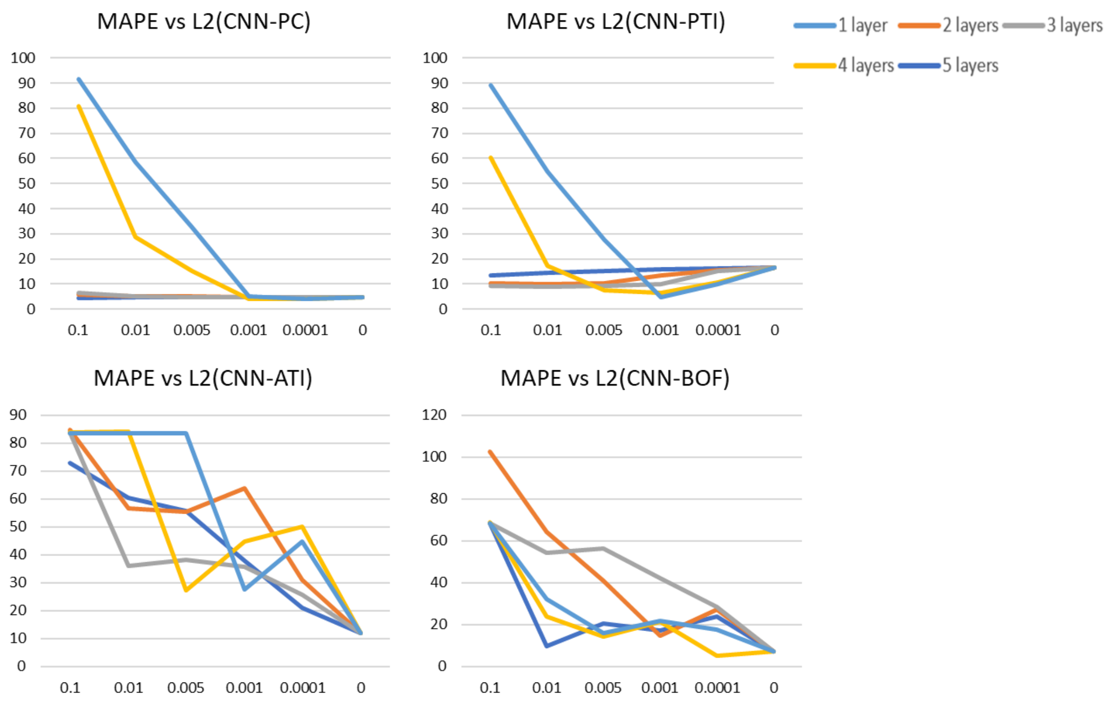
Experiment 19(L2 Regularization)



Appendices Figure 16: MAPE vs L2(LSTM) for Experiment 19

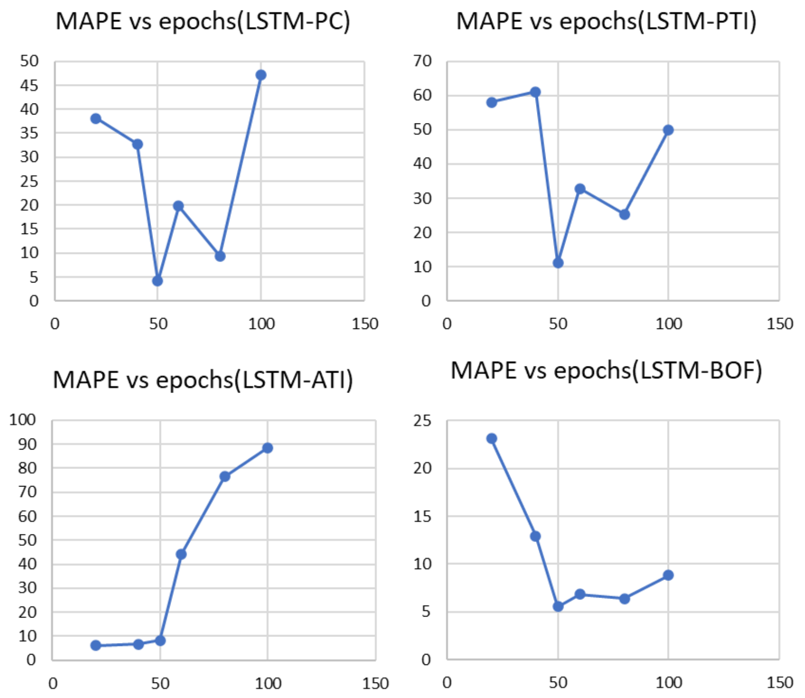


Appendices Figure 17: MAPE vs L2(GRUs) for Experiment 19

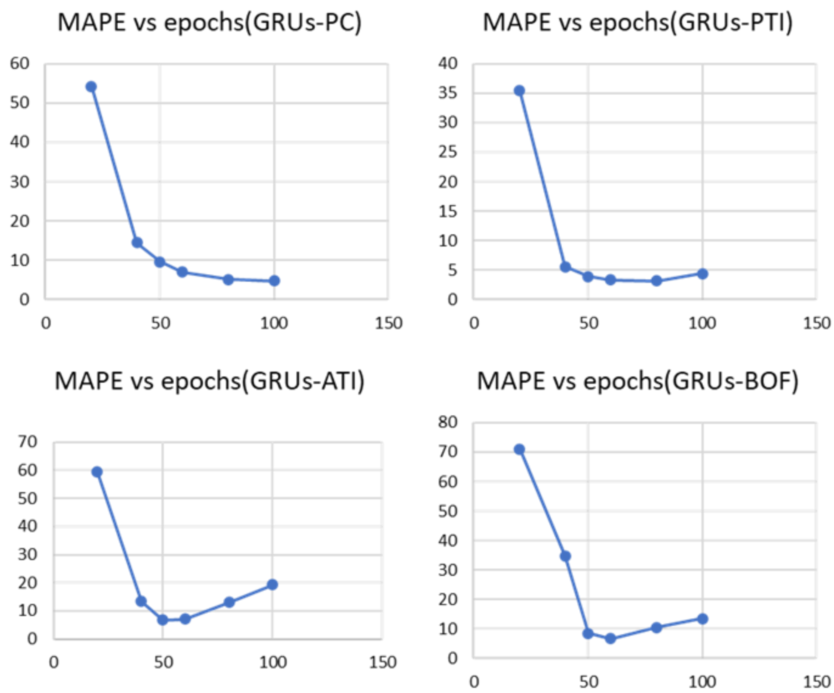


Appendices Figure 18: MAPE vs L2(CNN) for Experiment 19

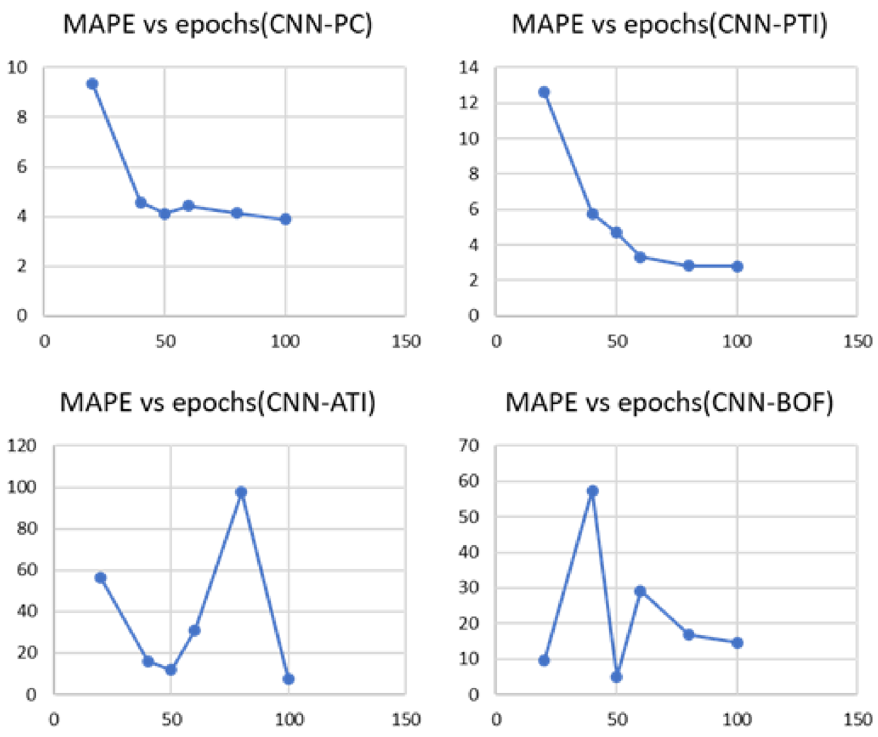
Experiment 20(Epochs)



Appendices Figure 19: MAPE vs Epochs(LSTM) for Experiment 20



Appendices Figure 20: MAPE vs Epochs(GRUs) for Experiment 20



Appendices Figure 21: MAPE vs Epochs(GRUs) for Experiment 20

Appendices D: Turnitin Report

fyp 2 thesis report bi19110220

ORIGINALITY REPORT

14%	10%	9%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	analyticsindiamag.com Internet Source	1%
2	Yash Indulkar. "Time Series Analysis of Cryptocurrencies Using Deep Learning & Fbprophet", 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021 Publication	1%
3	www.researchgate.net Internet Source	1%
4	medium.com Internet Source	1%
5	publikationen.bibliothek.kit.edu Internet Source	<1%
6	web.archive.org Internet Source	<1%
7	Mohil Maheshkumar Patel, Sudeep Tanwar, Rajesh Gupta, Neeraj Kumar. "A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions",	<1%