**METAVERSE-BASED CRYPTOCURRENCIES**

**PREDICTION USING MACHINE LEARNING**

**LO GUAN SIANG**

**FACULTY OF COMPUTING AND INFORMATICS UNIVERSITI MALAYSIA SABAH**

**2022**

**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 :** 16/6/22

**CERTIFIED BY;**

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.



16 JUN 2022

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

# 16 JUN 2022

# 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 preprocessing, 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 paper, we proposed a deep learning method based on Convolutional neural networks (CNNs), Long short-term memory (LSTM), and Gated recurrent units (GRUs) to predict the Smooth Love Potion (SLP), Sandbox (SAND) and Decentraland (MANA) closing price. We have achieved the MAPE of 25.2571 and 17.5064 by using the features of 'Open', 'High', 'Low', and 'Volume' to predict the closing price of SAND and MANA using the GRUs and LSTM model. For the SLP cryptocurrency, the best preliminary results we achieved are using 'Pre\_Close' as an input feature, and the MAPE is recorded as 167.2377 using the CNN model.

# *ABSTRAK*

# *RAMALAN MATA WANG KRIPTO BERASASKAN METAVERSE DENGAN MENGGUNAKAN PEMBELAJARAN MESIN.*

*Mata wang kripto baru-baru ini telah menarik banyak minat daripada pelabur dan penyelidik. Mata wang kripto telah menjadi fenomena global dalam sektor kewangan dan pilihan pilihan pelabur untuk instrumen kewangan yang didagangkan kerana kesederhanaan, inovasi, keselamatan dan ketelusannya, serta sifatnya yang terdesentralisasi. Mata wang kripto berasaskan Metaverse ialah satu subtopik kes yang digunakan untuk mata wang kripto yang digunakan untuk perdagangan dan pelaburan dalam aset digital dan tanah di metaverse, dunia maya yang merupakan lanjutan daripada dunia nyata. Mata wang kripto ini, termasuk yang berasaskan metaverse, mempunyai turun naik yang drastik dan pergerakan agresif dalam harga mereka, yang sangat tidak dapat diramalkan. Walaupun terdapat beberapa kajian terbaru untuk pelaksanaan pembelajaran mesin untuk meramalkan mata wang kripto, ia tidak pernah dilaksanakan dalam mata wang kripto berasaskan Metaverse. Kajian ini akan mereka bentuk model pembelajaran mesin untuk ramalan harga mata wang kripto berasaskan metaverse. Tambahan pula, banyak algoritma pembelajaran mesin yang diselia akan dilaksanakan sebagai model ramalan untuk harga penutupan mata wang kripto berasaskan metaverse. Pembangunan model pembelajaran mesin akan mengikuti proses seperti di bawah: pengumpulan data, prapemprosesan data, pemilihan model, latihan dan pembangunan model dan penilaian modal. Akhir sekali, sumbangan kajian ini boleh disimpulkan seperti berikut. Kajian ini bertujuan untuk membantu pelabur dan penyelidik untuk meminimumkan risiko dalam pasaran mata wang kripto dan mempelbagaikan pengurusan portfolio mata wang kripto. Di samping itu, dari perspektif pemain metaverse, kajian ini bertujuan untuk menilai sama ada bermain secara berterusan untuk permainan play-to-ear untuk mendapatkan mata wang kripto berasaskan metaverse adalah balasan dengan pulangan positif atau tidak. Dari perspektif syarikat, ramalan optimis mata wang kripto berasaskan metaverse menggalakkan mereka untuk membangunkan lebih banyak projek metaverse. Dalam kertas kerja ini, kami mencadangkan kaedah pembelajaran mendalam berdasarkan rangkaian neural Konvolusi (CNN), Memori jangka pendek panjang (LSTM) dan unit berulang berpagar (GRU) untuk meramalkan Smooth Love Potion (SLP), Sandbox (SAND) dan Harga penutupan Decentraland (MANA). Kami telah mencapai MAPE 25.2571 dan 17.5064 dengan menggunakan ciri 'Open', 'High', ‘Low’ dan 'Volume' untuk meramalkan harga penutup SAND dan MANA menggunakan model GRU dan LSTM. Untuk mata wang kripto SLP, keputusan awal terbaik yang kami capai adalah menggunakan 'Pre\_Close' sebagai ciri input, dan MAPE direkodkan sebagai 167.2377 menggunakan model CNN.*

**TABLE OF CONTENTS**

# 

**TITLE**

Page

[**DECLARATION**](#_1fob9te) ii

[**ACKNOWLEDGEMENT**](#_3znysh7) iii

[**ABSTRACT**](#_tyjcwt) iv-v

[***ABSTRAK***](#_1t3h5sf) vi-vii

[**TABLE OF CONTENTS**](#_4d34og8) viii-x

[**LIST OF TABLES**](#_3rdcrjn) xi

[**LIST OF FIGURES**](#_35nkun2) xii-xiii

[**LIST OF APPENDICES**](#_35nkun2) xiv

[**CHAPTER 1: INTRODUCTION**](#_1ksv4uv) 1

[1.1 Introduction](#_2jxsxqh) 1-3

[1.2 Problem Background](#_z337ya) 3-5

[1.3 Problem Statement](#_147n2zr) 5

[1.4 Project Objective (PO)](#_3o7alnk) 6

[1.5 Project Scope](#_3j2qqm3) 6

[1.6 Organization of the Report](#_4i7ojhp) 7

[**CHAPTER 2: LITERATURE REVIEW**](#_2xcytpi) 8

[2.1 Introduction](#_1ci93xb) 8

[2.2 Methodology of Literature Review](#_3whwml4) 9-11

[2.3 Classification of the Reviewed Literature](#_2bn6wsx) 11

[2.3.1 By Country](#_32hioqz) 11-12

[2.3.2 By Year](#_32hioqz) 12

[2.3.3 Source Based](#_32hioqz) 13

[2.4 Brief Review of Cryptocurrency Prediction Techniques](#_ihv636) 13

[2.4.1 Machine Learning Techniques](#_32hioqz) 14-25

[2.4.2 Return-predictive Features](#_1hmsyys) 25

[2.4.3 Interval of Prediction](#_1hmsyys) 26

[2.4.4 Type of Cryptocurrencies Predicted](#_1hmsyys) 26

2.5 Critical Summary 33-34

2.6 Conclusion34

[**CHAPTER 3: METHODOLOGY**](#_19c6y18) 35

[3.1 I](#_3tbugp1)ntroduction 35

[3.2 I](#_3tbugp1)mplementation Environment 35

[3.3 Data Collection](#_3tbugp1) 36-38

[3.4 Data Preprocessing](#_28h4qwu) 38-39

[3.5 M](#_nmf14n)odel Choosing 39

[3.5.1 Long Short-Term Memory](#_37m2jsg) 39-40

[3.5.2 Gated Recurrent Unit](#_1mrcu09) 40-41

[3.5.3 Convolutional Neural Networks](#_1mrcu09) 41-43

[3.6 Model Training and Development](#_46r0co2) 43-44

[3.7 Model Evaluation](#_46r0co2) 44-45

[3.8 Overall Flow of Activities](#_46r0co2) 45

[3.9 C](#_46r0co2)onclusion45

[**CHAPTER 4: EXPERIMENTAL DESIGN**](#_19c6y18) 46

4.1 Introduction 46-47

4.2 Data and Feature Engineering 47-55

4.3 Model Design 55

[4.3.1 Long Short-Term Memory](#_37m2jsg) Model 56-57

[4.3.2 Gated Recurrent Unit](#_1mrcu09) Model 57

[4.3.3 Convolutional Neural Networks](#_1mrcu09) Model 58

[4.4 Model Evaluation](#_46r0co2) 59

[4.5 Experiments Setup](#_46r0co2) 59

[4.5.1 Experiments 1 Setup](#_37m2jsg) 60-61

[4.5.2 Experiments 2 Setup](#_1mrcu09) 61-63

[4.5.3 Experiments 3 Setup](#_37m2jsg) 63-64

[4.6 C](#_46r0co2)onclusion65

[**CHAPTER 5: IMPLEMENTATION**](#_19c6y18) 66

[5.1 I](#_46r0co2)ntroduction66

[5.2 Experiment 1 Implementation](#_46r0co2) 66-67

[5.2.1 The Proposed LSTM Model](#_37m2jsg) 67-72

[5.2.2 The Proposed GRUs Model](#_1mrcu09) 72-74

[5.2.3 The Proposed CNN Model](#_37m2jsg) 74-76

[5.3 Experiment 2 Implementation](#_46r0co2) 76

[5.3.1 The Proposed LSTM model](#_37m2jsg) 76-78

[5.3.2 The Proposed GRUs model](#_1mrcu09) 78-79

[5.3.3 The Proposed CNN model](#_37m2jsg) 80-81

[5.4 Experiment 3 Implementation](#_46r0co2) 82

[5.4.1 The Proposed LSTM Model](#_37m2jsg) 82

[5.4.2 The proposed GRUs Model](#_1mrcu09) 82

[5.4.3 The proposed CNN Model](#_1mrcu09) 82-83

[5.5](#_46r0co2) Conclusion83

[**CHAPTER 6: PRELIMINARY RESULTS**](#_19c6y18) 84

[6.1 I](#_46r0co2)ntroduction84

[6.2 Experiment 1 Results and Analysis](#_46r0co2) 84

[6.2.1 SAND](#_37m2jsg) 84-85

[6.2.2 SLP](#_1mrcu09) 85

[6.2.3 M](#_37m2jsg)ANA 85-86

[6.3 Experiment 2 Results and Analysis](#_46r0co2) 86

[6.3.1 SAND](#_37m2jsg) 86

[6.3.2 SLP](#_1mrcu09) 87

[6.3.3 M](#_37m2jsg)ANA 87

[6.4 Experiment 3 Results and Analysis](#_46r0co2) 88

[6.4.1 SAND](#_37m2jsg) 88

[6.4.2 SLP](#_1mrcu09) 88-89

[6.4.3](#_1mrcu09) MANA 89

[6.5](#_46r0co2) Conclusion89-90

[**CHAPTER 7: C**](#_19c6y18)**ONCLUSION**91-92

[**REFERENCES**](#_3as4poj) 93-102

[**APPENDICES**](#_49x2ik5) 103

# 

# LIST OF TABLES

# 

Page

[Table 1 : Summary of Literature review 27-32](#_Toc6616799)

[Table 2 : Type of Hardware Requirement 3](#_Toc6616799)5

[Table 3 : Feature of the Metaverse-based Cryptocurrencies 37](#_Toc6616799)

[Table 4 : Parameter Value of LSTM, GRUs and CNN models 5](#_Toc6616799)5

[Table 5 : Results of SAND Prediction in Experiment 1 8](#_Toc6616799)4

[Table 6 : Results of SLP Prediction in Experiment 1 8](#_Toc6616799)5

[Table 7 : Results of MANA Prediction in Experiment 1 8](#_Toc6616799)5

[Table 8 : Results of SAND Prediction in Experiment 2 8](#_Toc6616799)6

[Table 9 : Results of SLP Prediction in Experiment 2 8](#_Toc6616799)7

[Table 10 : Results of MANA Prediction in Experiment 2 8](#_Toc6616799)7

[Table 11 : Results of SAND Prediction in Experiment 3](#_Toc6616799) 88

[Table 12 : Results of SLP Prediction in Experiment 3 8](#_Toc6616799)8

[Table 13 : Results of MANA Prediction in Experiment 3 8](#_Toc6616799)9

[Table 14 : Objectives Versus Progress FYP 1 9](#_Toc6616799)1

# LIST OF FIGURES

Page

[Figure 1: Overview of Cryptocurrency Prediction System](#_1rvwp1q)s[.](#_1rvwp1q) 9

[Figure 2: Classification of Studies on Cryptocurrency Market Forecasting by Country](#_1rvwp1q) 12

[Figure 3: Year-wise Categorization of Studies on the Cryptocurrency Market.](#_1rvwp1q) 12

[Figure 4: Distribution of Papers According to the Journals..](#_1rvwp1q) 13

[Figure 5: LSTM Cell Structure(Patel et al.).](#_1rvwp1q) 39

[Figure 6: Structure of a GRU(Patel et al.).](#_1rvwp1q) 41

[Figure 7: Architecture of the CNN.](#_1rvwp1q) 41

[Figure 8: Convolution-Layer.](#_1rvwp1q) 42

[Figure 9: Pooling-Layer.](#_1rvwp1q) 43

Figure 10: Fully Connected-Layer[.](#_1rvwp1q) 43

Figure 11: Overall Flow of Activities45

Figure 12: SAND Dataset[.](#_1rvwp1q) 47

Figure 13: SLP Dataset48

[Figure 14: MANA Dataset](#_1rvwp1q) 48

Figure 15: Null Values Inside the SAND Dataset51

Figure 16: Final SAND Dataset[.](#_1rvwp1q) 52

Figure 17: Final SLP Dataset[.](#_1rvwp1q) 52

Figure 18: Final MANA Dataset53

Figure 19: Dataset Bifurcation for SAND into Training and Validation54

Figure 20: Dataset Bifurcation for SLP into Training and Validation54

Figure 21: Dataset Bifurcation for SLP into Training and Validation55

Figure 22: The Architecture of LSTM Model56

Figure 23: .The Architecture of GRUs Model57

Figure 24:.The Architecture of CNN Model58

Figure 25: System Architecture59

Figure 26: Experiments 1 Setup for SAND60

Figure 27: Experiments 1 Setup for MANA61

Figure 28: Experiments 1 Setup for SLP61

Figure 29: Experiments 2 Setup for SAND61

Figure 30: Experiments 2 Setup for SLP 62

Figure 31: Experiments 2 Setup for MANA63

Figure 32: Experiments 3 Setup for SAND63

Figure 33: Experiments 3 Setup for SLP64

Figure 34: Experiments 3 Setup for MANA64

Figure 35: Code Snippet for Importing Library67

Figure 36: Code Snippet for Importing Dataset67

Figure 37: Code Snippet for Feature Engineering68

Figure 38: Code Snippet for Data Preprocessing68

Figure 39: Code Snippet for Feature Selection69

Figure 40: Code Snippet for Train-validation Split 69

Figure 41: Code Snippet for Prepare Train and Validation Data70

Figure 42: Build and Compile the LSTM Model71

Figure 43: Code Snippet for Using LSTM Model to Predict Closing Price 71

Figure 44: Code Snippet for Model Evaluation72

Figure 45: Build and Compile the GRUs Model73

Figure 46: Code Snippet for Using GRU Model to Predict Closing Price73

Figure 47: Build and Compile the CNN Model 75

Figure 48: Code Snippet for Using GRU Model to Predict Closing Price75

Figure 49: Code Snippet for LSTM Model Implementations for Experiment 277

Figure 50: Code Snippet for GRUs Model Implementations for Experiment 2 79

Figure 51: Code Snippet for CNN Model Implementations for Experiment 281

**LIST OF APPENDICES**

Page

Appendices A: Proposal Revision/Progress Revision 103-104

Appendices B: Meeting Log Screenshot105

Appendices C: Turnitin Report106

**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 LSTU 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 (PO)**

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 preprocessing, model choosing, model training and development, and model evaluation. Chapter 4 describes the data and feature engineering used in experiments 1, 2, and 3, 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 three experiments. Chapter 5 describes the python code implementation of experiments 1, 2 and 3 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, 2 and 3 to three metaverse cryptocurrencies: SAND, SLP and MANA, in the metrics of MSE, MAE, RMSE, and MAPE. Chapter 7 summarizes achievements that have been achieved so far and future work that can be done in Fyp 2.

**CHAPTER 2**

**LITERATURE REVIEW**

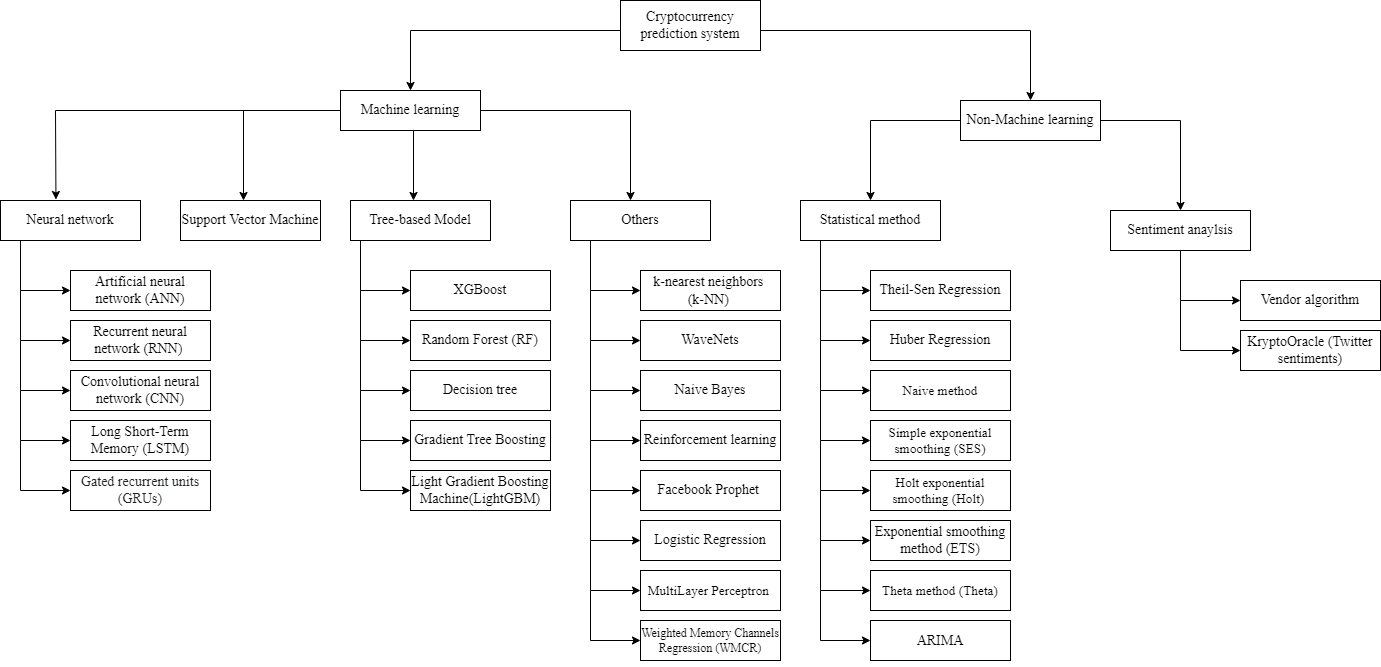
## **2.1 Introduction**

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 our literature review, we present a summary of recent studies on using machine learning and deep learning to predict the cryptocurrency's price. As a result, we can identify common approaches, analysis types, and findings. As a result, we evaluate the present state of study on this topic and future research possibilities. As a result, we provide a three-fold contribution. First, we give scholars in this field a summary of previous work, identifying recurring trends and unfilled niches. Second, based on the reviewed literature, we determine which promising strategies to solve the cryptocurrency price problem. Third, to improve transparency and speed scientific development, we set reporting guidelines for future research.

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**

Our literature search is based on the recommendation of Webster and Watson (2002) and vom Brocke et al. (2009). We began by examining many interdisciplinary research databases to establish our initial literature base (i.e., ACM Digital Library, Emerald insight, IEEE, Springerlink, ScienceDirect, Scopus). Then, the literature search was conducted by submitting the Google Scholar, Scopus, and IEEE databases with basic machine learning and cryptocurrency keywords. The scope of the search topic was limited to the recent few years (>2017). Submitted queries were 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.

This results in a preliminary list of 66 publications to be reviewed further. First, we eliminate 51 papers based on their titles and abstracts, as they do not expressly meet the subject of our literature review. This could be due to papers implementing methods that do not meet Gu et al.'s (2019) definition of machine learning, papers that do not focus on cryptocurrency price and return prediction, papers that are not available in English, or papers not using a prediction task, paper that only focus on the statistical method and no regarding the machine learning scope. Following that, a forward and backward search is implemented for the remaining relevant papers and returns another 15 articles, bringing the total number of publications for an in-depth review to 30.

First, the classification of the reviewed literature will be scrutinised and categorised by country of origin, year of publication, and source of literature obtained. Then, to further analyse the methodology and approaches used by the reviewed literature, we establish fundamental concepts for categorising the price prediction techniques within all review literature. We went over all of these publications and came up with an initial set of classification concepts. We reviewed these early notions throughout the paper screening process and adapted them as needed. Following that, all of the identified concepts were discussed and synthesised, yielding a final set of categorisation concepts:

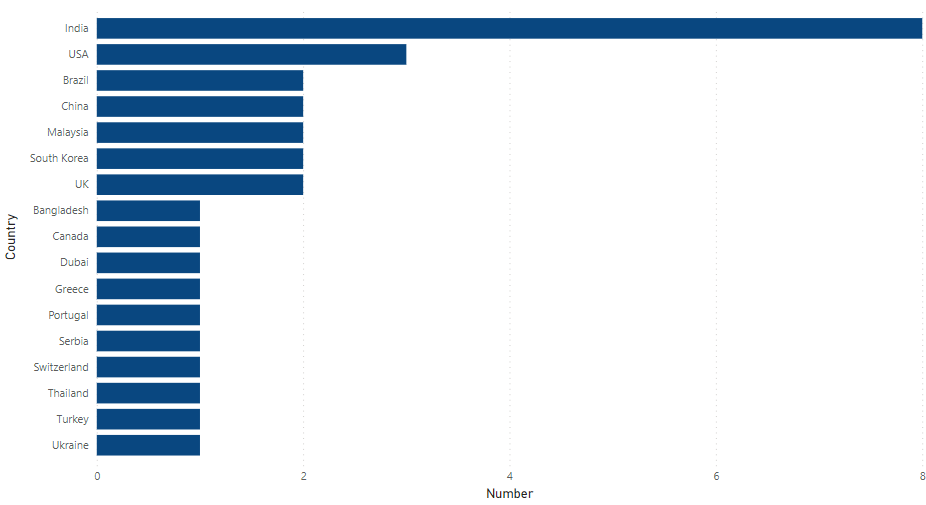
* Machine learning (Recurrent Neural networks method, Support Vector Machine method, SVM method, Tree-based method and Others)
* Features (i.e., technical-base, sentiment-based, asset-based and blockchain-based)
* Predictive intervals (i.e., second, minute, hour, day, week)
* Cryptocurrency type (i.e., 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**

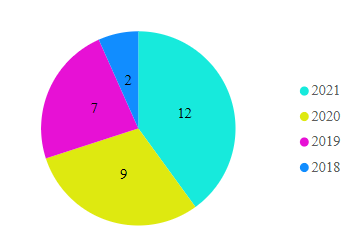
Figure 2 below illustrates the distribution of reviewed articles according to the country in where the study was undertaken. This area piqued the interest of researchers from a wide variety of countries. We analysed research articles published between 2018 and 2022 in 18 different nations. India submitted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



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

**2.3.3 By Year**

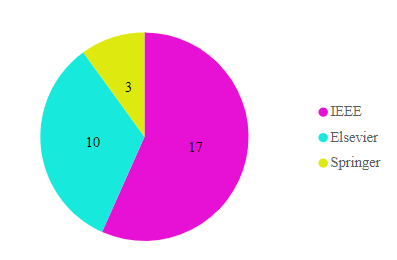
Figure 3 illustrates the year-by-year evaluation of the papers. According to the articles we picked, we discovered that studies on the cryptocurrency market have surged in recent years. The results demonstrate that research interest in this topic has increased. All peer-reviewed publications occurred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



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

**2.3.4 Source Based**

We chose relevant papers for cryptocurrency price prediction that received at least one citation published between 2018 and 2021 on Elsevier, Springer, or IEEE Xplore. In Figure 4, we can examine the distribution of publications by the journal. 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, we employ four main concepts: predictive features, prediction intervals, machine learning approaches, type of cryptocurrencies forecasted (Jaquart et al., 2020). These principles are pretty 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.

1. **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.

1. **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 (R2) 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%.

1. **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.

1. **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.

1. **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  Ripple  Bitcoin Cash  EOS  Litecoin  Cardano  Stellar  Tron | 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 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)  Convolutional neural network (CNN)  Random Forest (RF)  k-nearest neighbors (k-NN) | 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. |
| 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  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) | 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. |
| 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  MultiLayer 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 unfavorable to the comparative study between the papers. On the other hand, this paper 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. We review the existing body of literature in this study branch using Webster and Watson's and von Brooke et al. standards. The literature review is organised 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. We discovered a lack of transparency and comparability throughout the reviewed literature, limiting our ability to validate and reproduce model results and trading approaches. In light of these difficulties, we advise that future researchers expose critical model configurations more organised manner, publish and upload their model and data to an open research repository, and benchmark their model against other published models.

**CHAPTER 3**

**METHODOLOGY**

**3.1 Introduction**

This chapter discusses the research methodologies used throughout the study, from data collecting to model evaluation. This research will thoroughly examine the many available systems for forecasting metaverse-based cryptocurrencies. This chapter is divided into six sections. The first section provides an overview of the chapter. Section 2 provides the hardware specification and implementation environment. Section 3 details the data collection procedure. Section 4 discusses the dataset's characteristics and data-preprocessing scheme. Section 5 describes the proposed machine learning models that this paper has to implement. The model evaluation schemes are described in section 7.

**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 | 512GB |
| 4 | Operating System | 64-bit Window 10 |
| 5 | Software and Tools | Google Colab |

Table 3 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 Preprocessing 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.

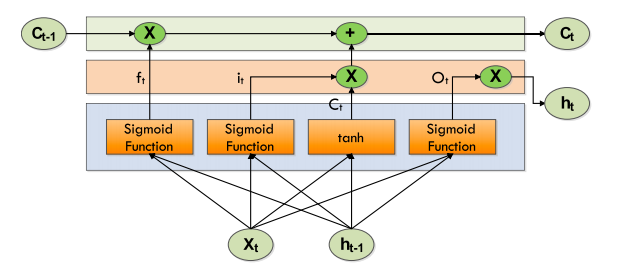
**3.5 Model Choosing**

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 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.

*it=σ* (*xt Vi + ht*-1 *Wi*)

*ft*=*σ* (*xt Vi* + *ht-*1 *Wi*)

*ot =σ* (*xtVo* + *ht*− 1*Wo*)

*= tanh* (*xtVg* + *ht*− 1*Wg*)

*Ct =σ* (*ft* ∗*Ct-*1+ it ∗*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), xt is input, ht− 1 is previous cell output, Ct− 1 is previous cell memory, ht is current cell output, Ct 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:

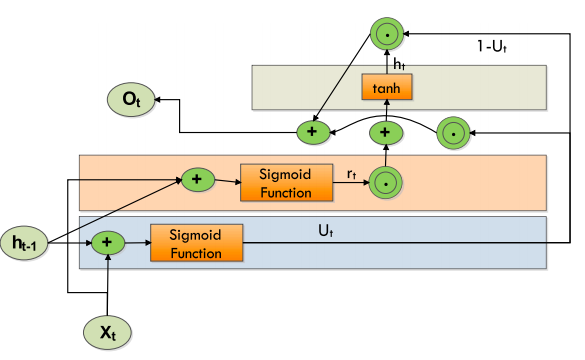
*ut = σ (Vuxt + Wuot− 1 + bu)*

*rt = σ (Vrxt + Wrot− 1 + br)*

*it = tanh (Voxt + Wo (rt ⊙ ot− 1) + bo)*

*ot = ut ⊙ ot− 1 + (1 − ut) ⊙ it*

Where xt is the input, ot is the output, ut is the update gate output, rt is the reset gate output, ⊙denotes the Hadamard product, and V, W, and b are the parameters or weight matrices.

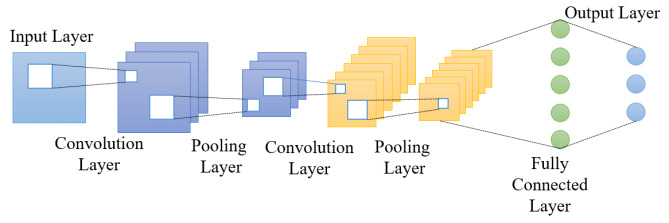


**Figure 6 : Structure of a GRU (Patel et al.)**

**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).

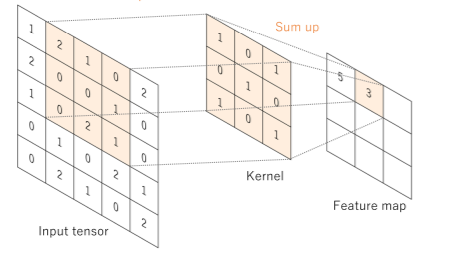
CNN comprises neurons or filters with weights, parameters, and biases that can be adjusted (Lu et al., 2021). Convolutional, pooling, and fully connected layers make up CNN's structure. The feature extraction part of the CNN is where the input from each neuron is associated with the previous layer's 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**

**A. Convolution Layer**

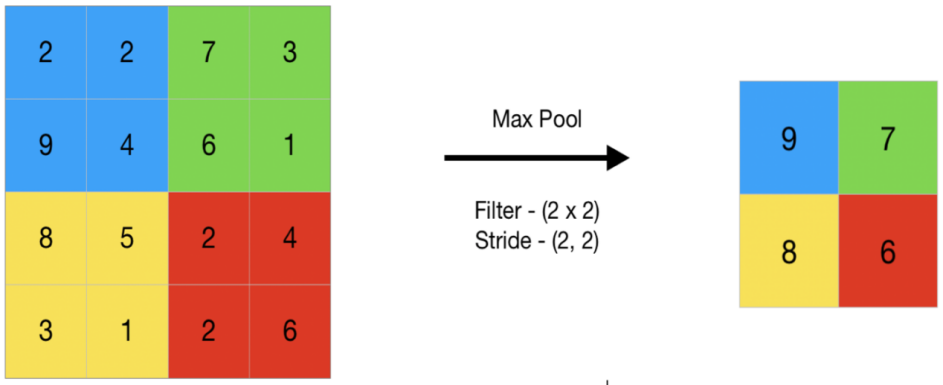
The convolutional layer is the central part of a Convolutional network that performs the most computational elevation. The goal of the convolution layer is to extract key features from image input data (Zhu et al., 2017). Convolution preserves the spatial relationship between pixels by learning image 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**

**B. Pooling Layer**

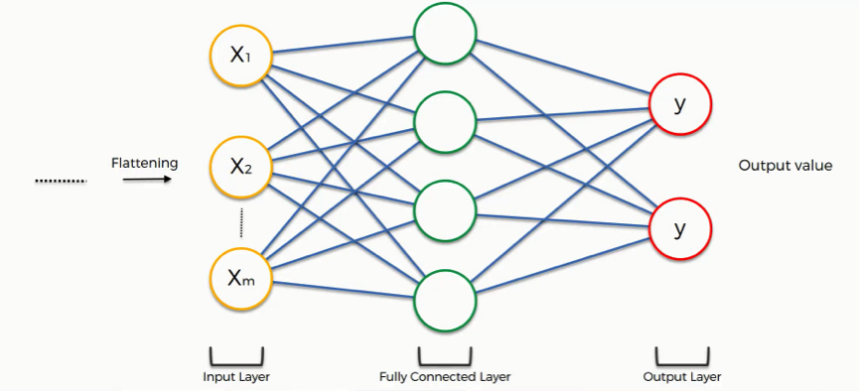
The pooling layer, as represented in Figure 9, decreases the dimension of each activation map while keeping the essential information. The input photos are divided into rectangles that do not overlap. Each area will be sampled by a nonlinear activity such as limit or average (Albawi et al., 2017). This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



**Figure 9: Pooling-Layer**

1. **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.

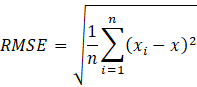
**3.7 Model Evaluation**

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.



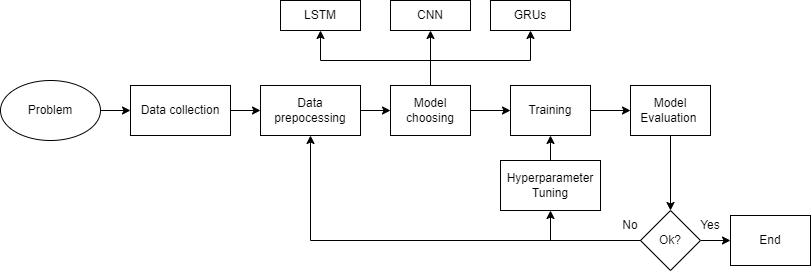






where Xi represents the predicted price, X represents the actual price and N is total number of observations.

**3.8 Overall Flow of Activities**



**Figure 11: Overall Flow of Activities**

**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 is 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 RNN 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.

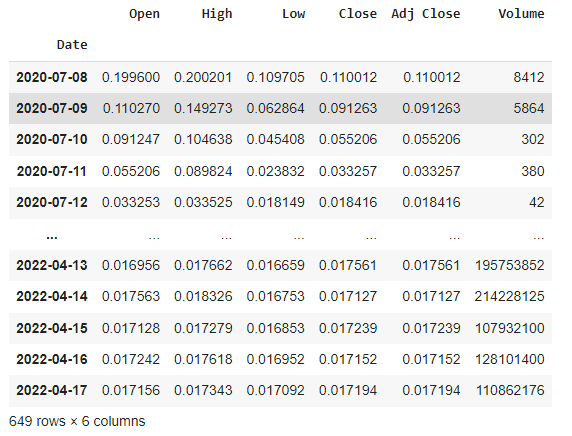
Furthermore, the data feature engineering is also discussed in this section about the algorithm involved in data and feature engineering for both experiments for SAND, SLP and MANA. The parameter and architecture of the three proposed models are also discussed in this section. At the end of this section,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



**Figure 12: SAND Dataset**



**Figure 13: SLP Dataset**



**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 7days (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:



Where An 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:

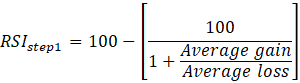


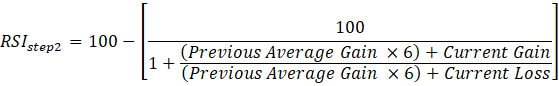
Where t is today, y is yesterday, N is the number of days in EMA, k=2÷(N+1).

​

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:

​



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 recent six closing prices to predict the next days day's relative strength index. Since the null values columns are comparatively small to the whole datasets. The ways are used to handle missing values are by deleting the rows or columns having null values.

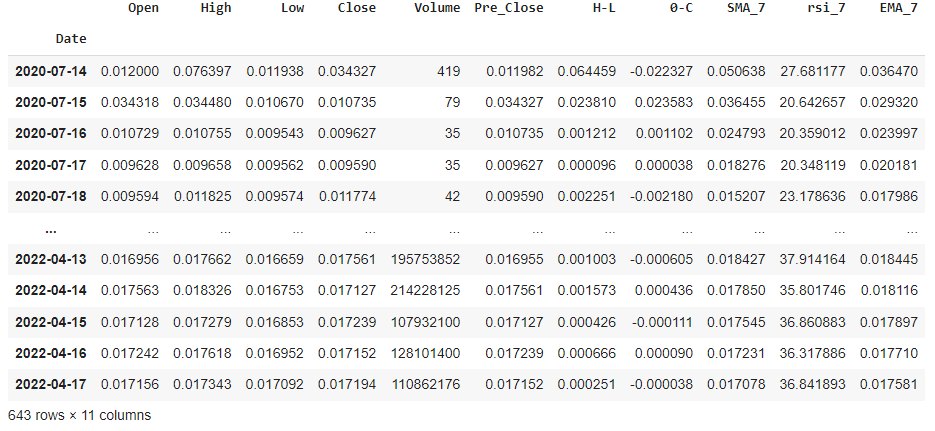


**Figure 15: Null values Inside the SAND Dataset**

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 4 starting 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.



**Figure 16: Final SAND Dataset**



**Figure 17: Final SLP Dataset**



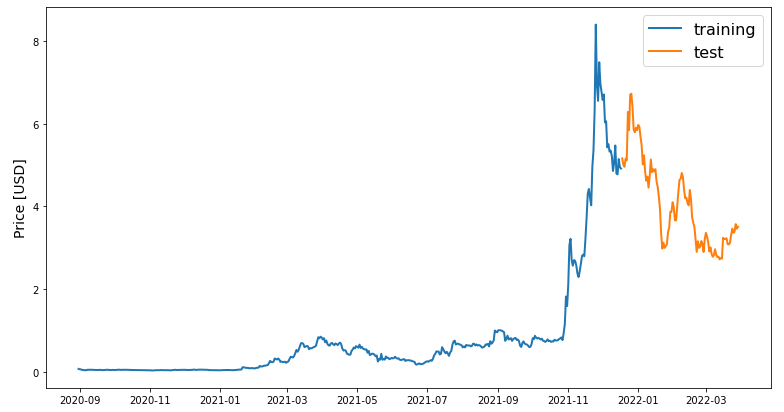
**Figure 18: Final MANA dataset**

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 experiments 1, 2, and 3, 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:

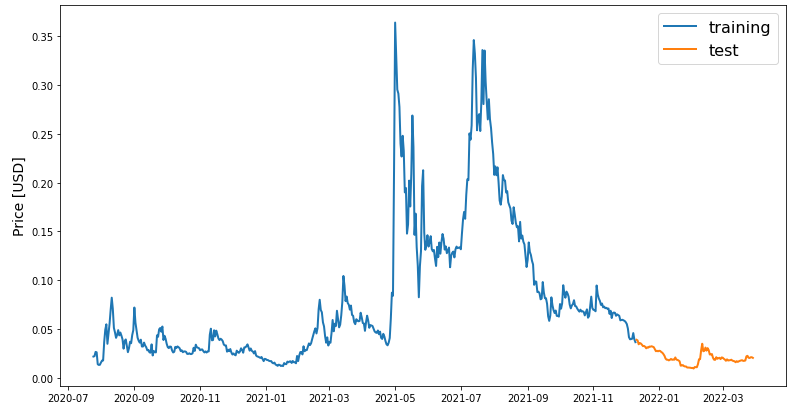


Where x is an original value, x’ is the normalized value.

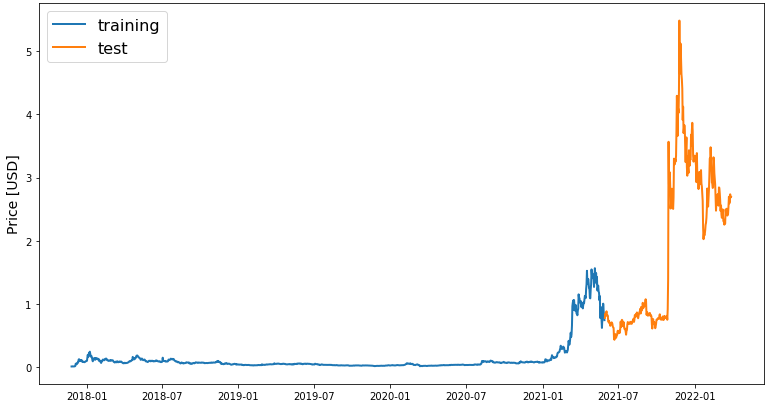
The dependent features used the target features were bifurcated into training & validation with an 80-20 ratio; 80% was for the training purpose, and the remaining 20 % was for validation purposes based on the length of the dataset. The predicted horizon is also the same for both three experiments, which 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**

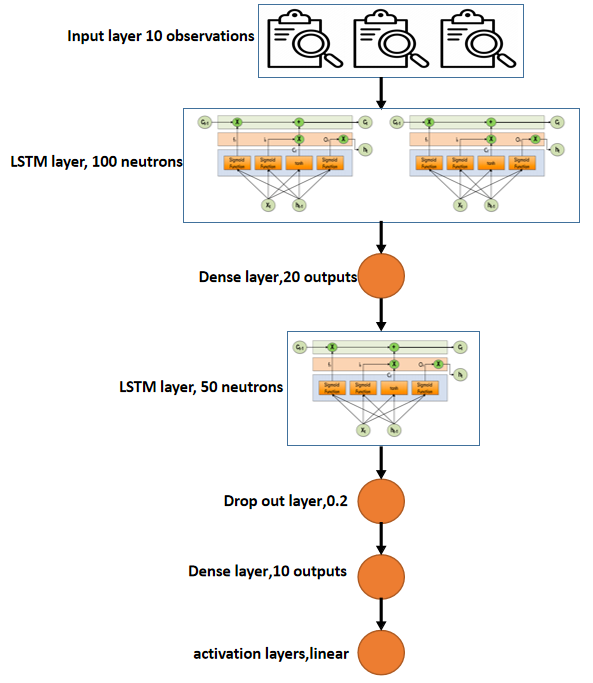
**4.3 Model Design**

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 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.

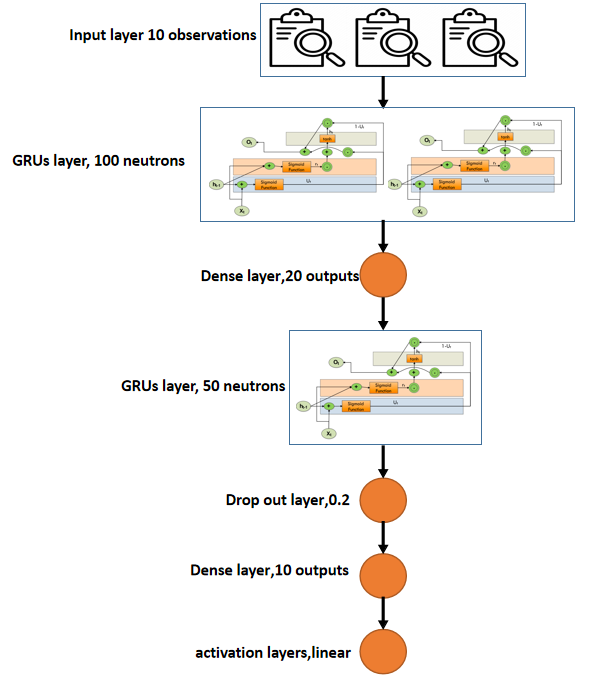






Where, the mt & vt are the values of vectors that are initialized to 0’s. Further, the biases in the optimizer can be seen in respectively,  is the learning rate.

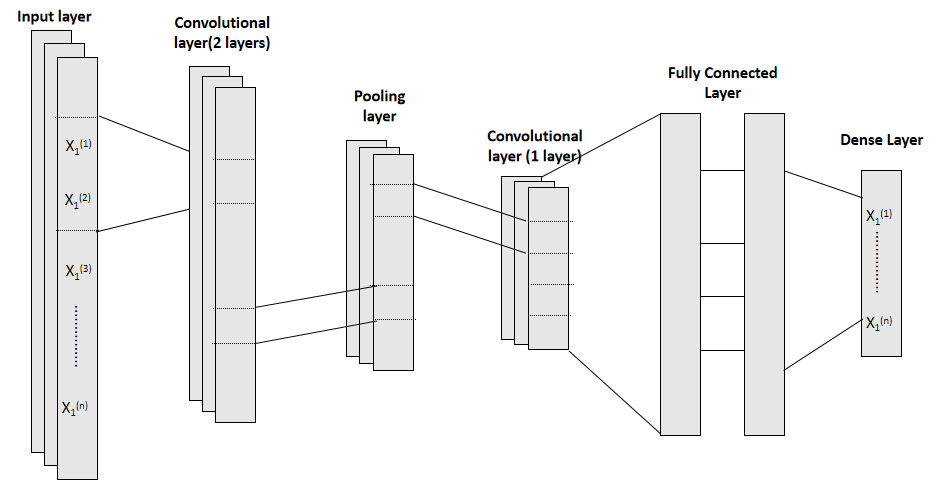
**4.3.2 GRUs model**



**Figure 23: The Architecture of GRUs Model**

Similarly, we construct the GRUs model that 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**

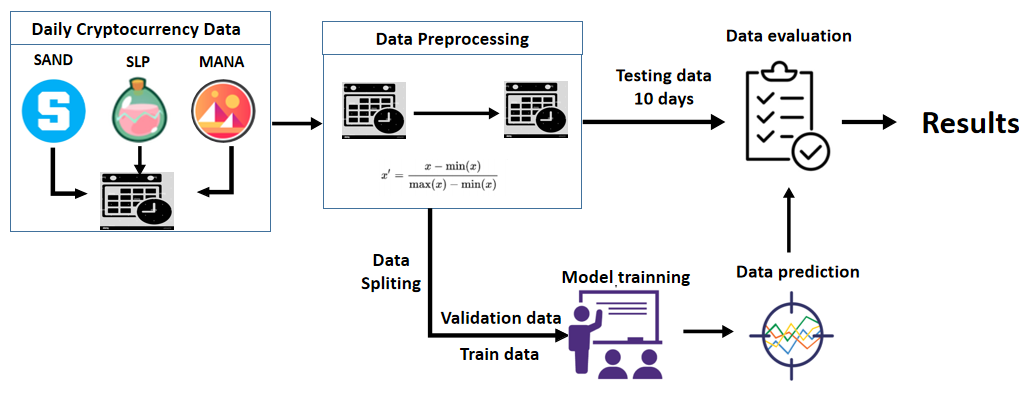


**Figure 24:The Architecture of CNN Model**

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 below:

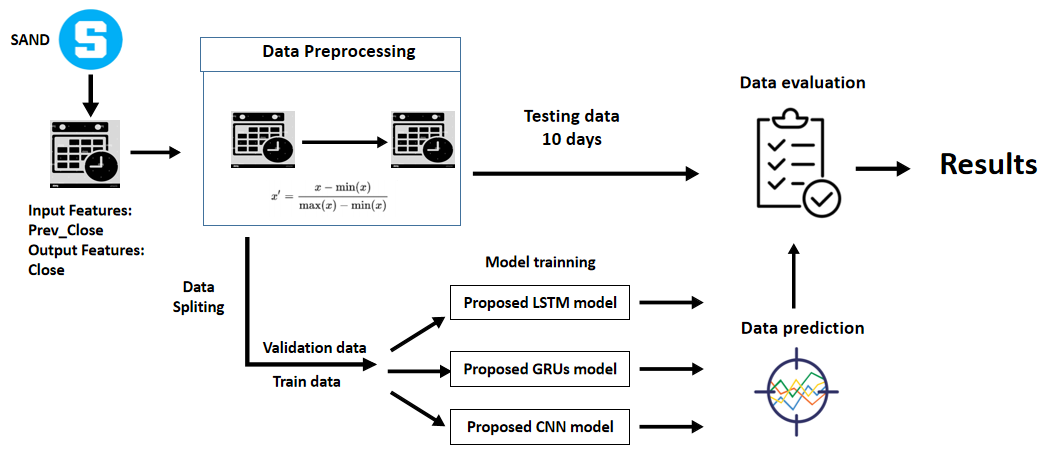


**Figure 25: System Architecture**

**4.5 Experiments Setup**

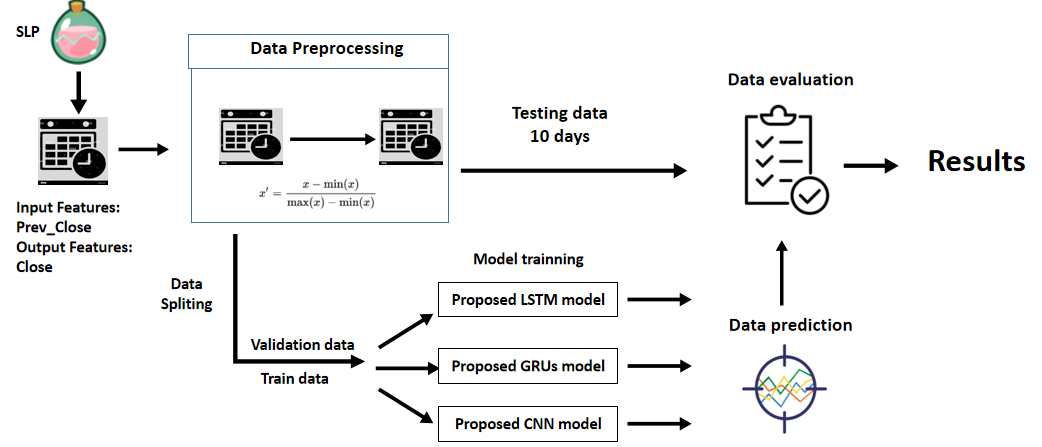
The experiments are setup up to compare the predictive power of the 11 dependent features, which are Open, High, Low, Close, Volume, Pre\_Close, H-L, O-C, SMA\_7, rsi\_7 and EMA\_7, towards the target and predictive variable. To check the predictive power of different features towards the target variables with three cryptocurrencies, SAND, SLP and MANA, the three experiments are set up with the various input variables.

**4.5.1 Experiments 1 Setup**

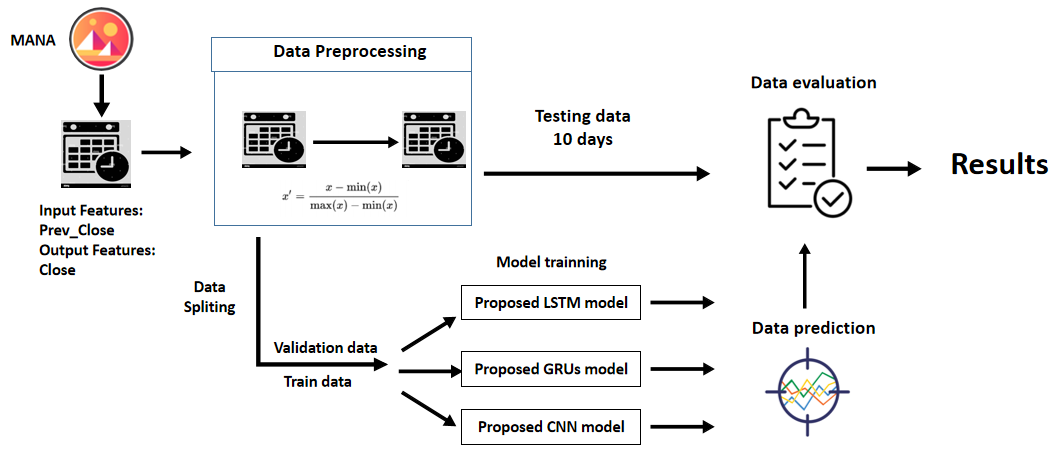


**Figure 26: Experiments 1 Setup for SAND**

Experiment 1 is set up like the diagram above 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 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 27 and 28 below.

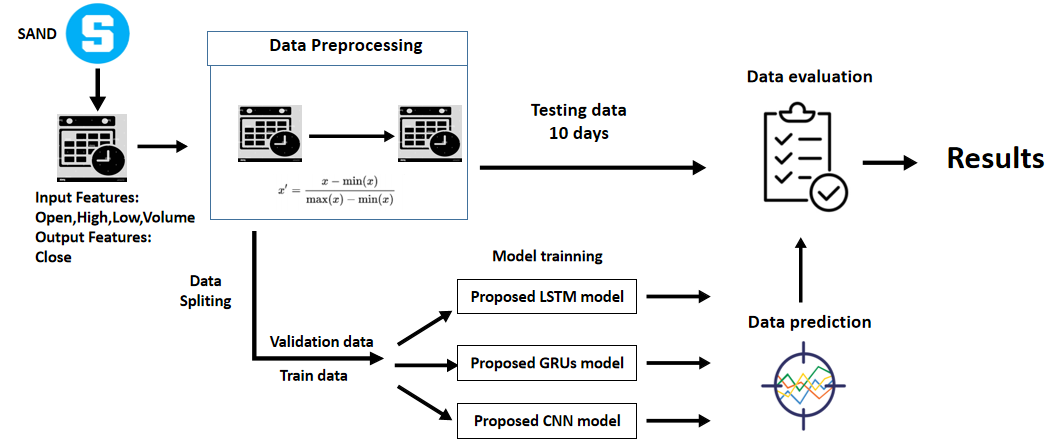


**Figure 27: Experiments 1 Setup for SLP**



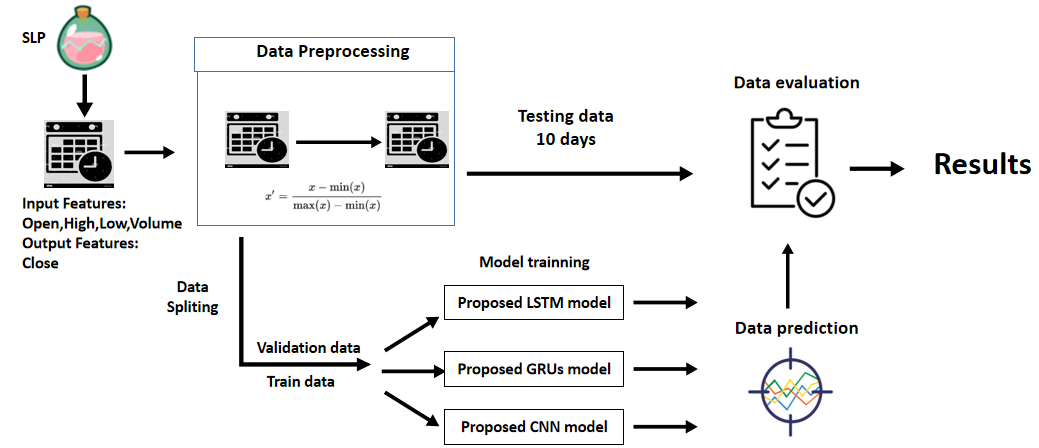
**Figure 28: Experiments 1 Setup for MANA**

**4.5.2 Experiments 2 Setup**

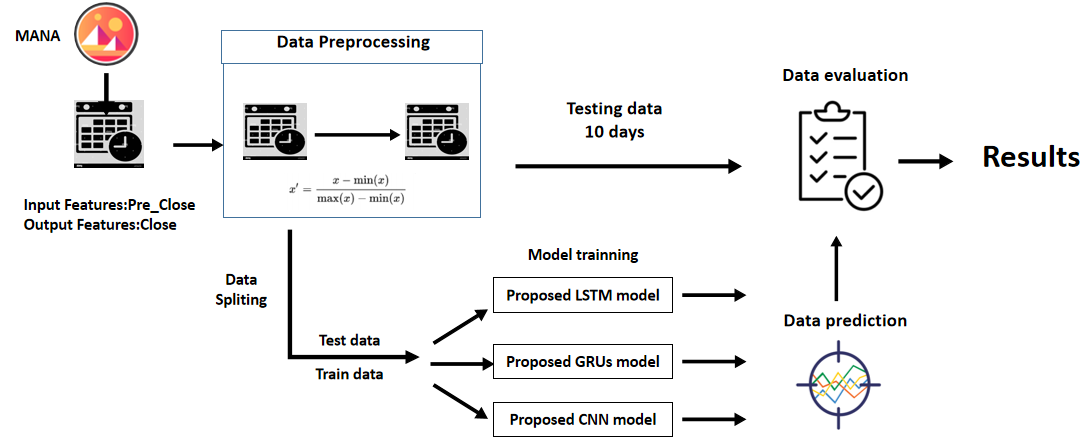


**Figure 29: Experiments 2 Setup for SAND**

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 Figures 30 and 31 below.

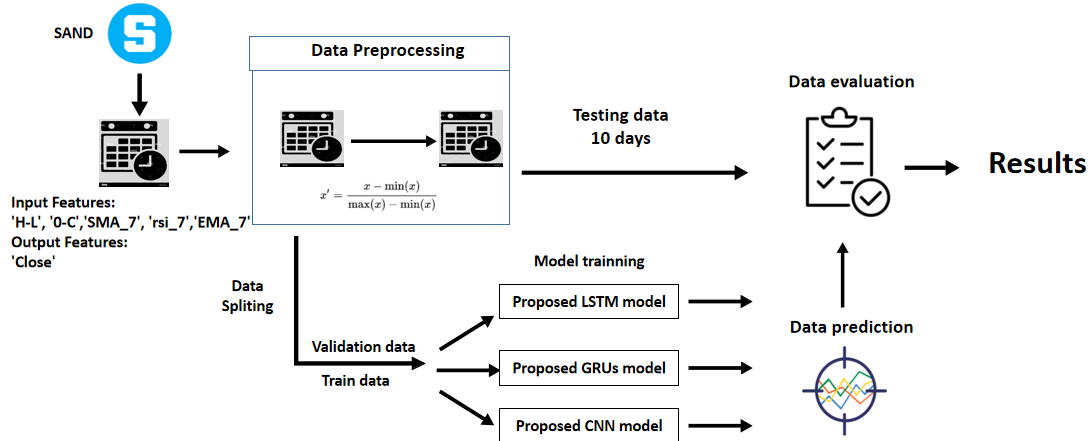


**Figure 30: Experiments 2 Setup for SLP**



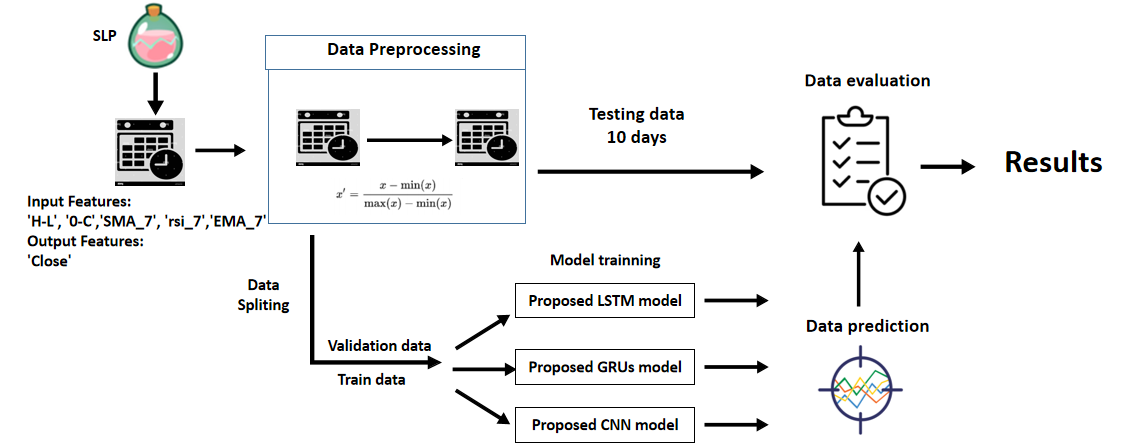
**Figure 31: Experiments 2 Setup for MANA**

**4.5.3 Experiments 3 Setup**

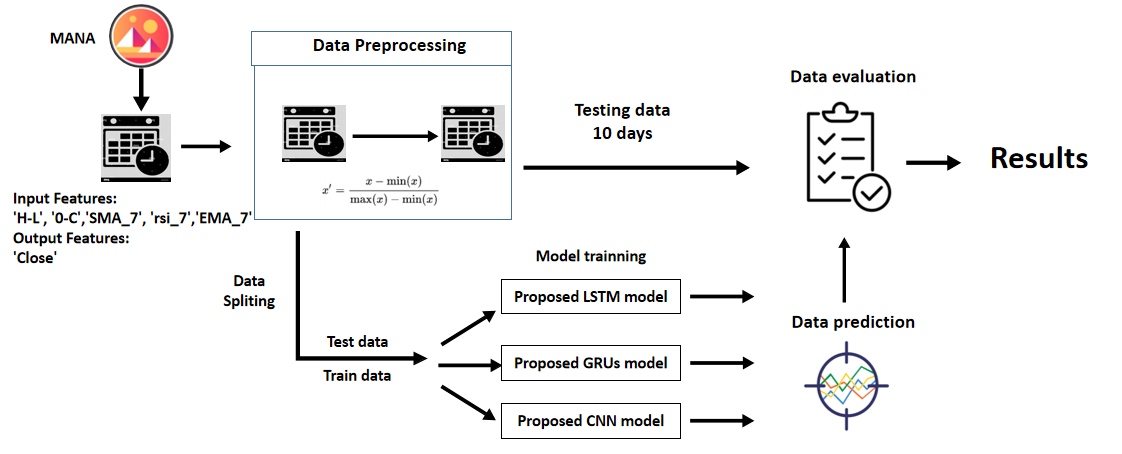


**Figure 32: Experiments 3 Setup for SAND**

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 Figures 33 and 34 below.



**Figure 33: Experiments 3 Setup for SLP**



**Figure 34: Experiments 3 Setup for MANA**

**4.6 Conclusion**

In conclusion, chapter 4, experimental design, describes and shows the flow of experiments 1,2 and 3 in diagram form. The experimental design is explained explicitly throughout the chapter, including data and feature engineering used in the experiments 1, 2 and 3, 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 section discusses the implementation of experiments 1, 2, and 3 by using the Google Colab as the implementation environment and Python 3 as the implementation language. 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.

**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 35: Code Snippet for Importing Library**

At first, we need to import the library that is useful for data preprocessing, 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, 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 36: Code Snippet for Importing Dataset**

The Sand, SLP and MANA datasets downloaded from Yahoo Finance in CSV format have been uploaded to the Github. We import the sand, SLP and mana dataset through the GitHub link. 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 37: Code Snippet for Feature Engineering**

The new data frame is generated by adding the 6 new features: '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 38: Code Snippet for Data Preprocessing**

This data\_preprocessing function 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 39: 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. 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 40: Code Snippet for Train-validation Split**

The look\_back and horizon are set as ten days, 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', 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 41: Code Snippet for Prepare Train and Validation Data**

In Figure 41, 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 42: Build and Compile the LSTM Model**

In Figure 42, 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 43: 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. 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 44: Code Snippet for Model Evaluation**

Finally, evaluate the result with standard performance metrics. 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 Figures 35 to 41. The steps include importing the library, dataset, feature engineering, data\_preprocessing, 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\_data,validation\_steps=50,verbose=1)

**Figure 45: 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 45. 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 layers, 0.2 dropout layers, 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)

pred\_Inverse

validate = (df[['Close']].tail(10))

timeseries\_evaluation\_metrics\_func(validate['Close'],pred\_Inverse[0])

**Figure 46: 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 46 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 (Figures 35 to 41). The steps include importing library, dataset, feature engineering, data preprocessing, 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.shape[-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\_data,validation\_steps=50,verbose=1)

**Figure 47: 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 47. The proposed CNN model consists of 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 the training of 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 48: 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. 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 48 for model evaluation. The evaluated results and analysis will be tabulated in table forms to compare with another two models, which are 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 which is 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 basically 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 Figure 49.

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 49: 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 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.

**5.3.2 The Proposed GRUs Model**

For the GRUs model, the experiment 2 flow is basically 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 Figure 50.

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)

pred\_Inverse

validate = (df[['Close']].tail(10))

timeseries\_evaluation\_metrics\_func(validate['Close'],pred\_Inverse[0])

**Figure 50: 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 basically 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 Figure 51.

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.shape[-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 51: 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 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.

**5.4 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 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 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 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 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 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 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 from 'Open', 'High', 'Low', 'Volume' to 'Pre\_Close', 'H-L', 'O-C', 'SMA\_7', 'rsi\_7', 'EMA\_7'.

**5.5 Conclusion**

In conclusion, chapter 5 describes the python code implementation of experiments 1,2 and 3 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 experiments 1, 2 and 3, model implementation of the proposed LSTM, GRUs and CNN model and the model evaluation metrics.

**CHAPTER 6**

**PRELIMINARY RESULTS**

**6.1 Introduction**

This section discusses the results of experiments 1,2 and 3 to three metaverse cryptocurrencies, which are SAND, SLP and MANA, in the metrics of MSE, MAE, RMSE, and MAPE. 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.

**6.2 Experiment 1 Results and Analysis**

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. Experiment 1 uses ‘Prev\_Close’ as an input feature to predict the next ten days of closing price, labelled as ‘Close’ target features.

**6.2.1 SAND**

**Table 5: Results of SAND Prediction in Experiment 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Metrics | | | |
| MSE | MAE | RMSE | MAPE |
| LSTM | 5.5132 | 2.3295 | 2.3480 | 82.4799 |
| GRUs | 0.8260 | 0.7819 | 0.9088 | 27.9069 |
| CNN | 1.2775 | 0.9015 | 1.1303 | 32.1328 |

In experiment 1, GRUs outperformed the other models by using ‘Prev\_Close’ as an input feature to predict the closing price of SAND metaverse cryptocurrency as the four metrics values record the lowest value. MAPE of the GRUs has been recorded as 27.9069 means the average of the absolute percentage errors of forecasts is 27.9069%. The other models, CNN and LSTM, are recorded as means absolute percentage errors of forecasts of 32.1328% and 82.4799%.

**6.2.2 SLP**

**Table 6: Results of SLP Prediction in Experiment 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Metrics | | | |
| MSE | MAE | RMSE | MAPE |
| LSTM | 0.0043 | 0.0627 | 0.0657 | 352.0169 |
| GRUs | 0.0020 | 0.0419 | 0.0441 | 236.1946 |
| CNN | 0.0016 | 0.0304 | 0.0436 | 167.2377 |

In experiment 1, CNN is comparatively better than the other models in predicting the closing price of SLP metaverse cryptocurrency as the four values of metrics record the lowest value. MAPE of the CNN is recorded as 167.2377 means the average of the absolute percentage errors of forecasts is 167.2377%. The other models, which are GRUs and LSTM, record as means absolute percentage errors of forecasts of 236.1946% and 352.0169%. We can see that both three algorithms are underperformed in predicting the SLP in experiment 1 as their MAPE is very high.

**6.2.3 MANA**

**Table 7: Results of MANA Prediction in Experiment 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Metrics | | | |
| MSE | MAE | RMSE | MAPE |
| LSTM | 0.8890 | 0.9392 | 0.9429 | 42.9515 |
| GRUs | 1.0524 | 1.0219 | 0.4678 | 46.7318 |
| CNN | 0.3532 | 0.5832 | 0.5943 | 26.7349 |

In experiment 1, CNN is comparatively better than the other models in predicting the closing price of MANA metaverse cryptocurrency as the four values of metrics record the lowest value. MAPE of the CNN is recorded as 26.7349 means the average of the absolute percentage errors of forecasts is 26.7349%. The other models, LSTM and GRUs, are recorded as means absolute percentage errors of forecasts of 42.9515% and 46.7318%.

**6.3 Experiment 2 Results and Analysis**

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.

**6.3.1 SAND**

**Table 8: Results of SAND Prediction in Experiment 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Metrics | | | |
| MSE | MAE | RMSE | MAPE |
| LSTM | 10.3318 | 3.1820 | 3.2143 | 112.8267 |
| GRUs | 0.8384 | 0.7150 | 0.9156 | 25.2571 |
| CNN | 5.6591 | 2.0351 | 1.1303 | 71.5376 |

In experiment 2, GRUs outperform the other models to predict the closing price of SAND metaverse cryptocurrency as the four values of metrics record the lowest value. MAPE of the GRUs has been recorded as 25.2571 means the average of the absolute percentage errors of forecasts is recorded as 25.2571%. The other models, LSTM and GRUs, are recorded as means absolute percentage errors of forecasts of 112.8267% and 71.5376%.

**6.3.2 SLP**

**Table 9: Results of SLP Prediction in Experiment 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Metrics | | | |
| MSE | MAE | RMSE | MAPE |
| LSTM | 0.0029 | 0.0540 | 0.0540 | 306.9561 |
| GRUs | 0.0307 | 0.1698 | 0.1754 | 974.9469 |
| CNN | 0.0174 | 0.1250 | 0.1318 | 718.1500 |

In experiment 2, LSTM is comparatively better than the other models for predicting the closing price of SLP metaverse cryptocurrency as the four values of metrics record lowest value. MAPE of the LSTM is recorded as 306.9561 means the average of the absolute percentage errors of forecasts is 306.9561%. The other models, GRUs and CNN, record as means absolute percentage errors of forecasts of 974.9469% and 718.1500%. We can see that both three algorithms are underperformed in predicting the SLP in experiment 2 as their MAPE is very high.

**6.3.3 MANA**

**Table 10: Results of MANA Prediction in Experiment 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Metrics | | | |
| MSE | MAE | RMSE | MAPE |
| LSTM | 0.1562 | 0.3839 | 0.3953 | 17.5064 |
| GRUs | 2.5512 | 1.5920 | 1.5972 | 72.8597 |
| CNN | 0.3569 | 0.5753 | 0.5975 | 26.3973 |

In experiment 2, LSTM is outperformed the other models to predicted the closing price of MANA metaverse cryptocurrency as the 4 values of metrics record the lowest value. MAPE of the LSTM has been recorded as 17.5064 means the average of the absolute percentage errors of forecasts is recorded as 17.5064%. The other models, GRUs and CNN, record as means absolute percentage errors of forecasts of 72.8597% and 26.3973%.

**6.4 Experiment 3 Results and Analysis**

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.

**6.4.1 SAND**

**Table 11: Results of SAND Prediction in Experiment 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Metrics | | | |
| MSE | MAE | RMSE | MAPE |
| LSTM | 1.0109 | 3.6440 | 3.6498 | 128.9782 |
| GRUs | 1.9047 | 0.8952 | 1.0054 | 31.2915 |
| CNN | 3.9226 | 1.8059 | 1.9806 | 64.1215 |

In experiment 3, GRUs outperformed than with the other models to predict the closing price of SAND metaverse cryptocurrency as the 4 values of metrics record the lowest value. MAPE of the GRUs has been recorded as 31.2915 means the average of the absolute percentage errors of forecasts is recorded as 31.2915%. The other models, which are LSTM and CNN, are recorded as means absolute percentage errors of forecasts of 128.9782% and 64.1215%.

**6.4.2 SLP**

**Table 12: Results of SLP Prediction in Experiment 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Metrics | | | |
| MSE | MAE | RMSE | MAPE |
| LSTM | 0.0087 | 0.0923 | 0.0930 | 522.1622 |
| GRUs | 0.0046 | 0.0662 | 0.0675 | 374.4992 |
| CNN | 0.0386 | 0.1932 | 0.1967 | 1089.6187 |

In experiment 3, GRUs is comparatively better than the other models for predicting the closing price of SLP metaverse cryptocurrency as the four values metrics record the lowest value. MAPE of the GRUs has been recorded as 374.4992 means the average of the absolute percentage errors of forecasts is recorded as 374.4992%. The other models, LSTM and CNN, record as means absolute percentage errors of forecasts of 522.1622% and 1089.6187%. We can see that both algorithms underperformed in predicting the SLP in experiment 3 as their MAPE are very high.

**6.4.3 MANA**

**Table 13: Results of MANA Prediction in Experiment 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Metrics | | | |
| MSE | MAE | RMSE | MAPE |
| LSTM | 2.1287 | 1.4563 | 1.4590 | 66.6440 |
| GRUs | 4.1785 | 2.0418 | 2.0441 | 93.4819 |
| CNN | 1.2789 | 1.1127 | 1.1308 | 50.8306 |

In experiment 3, CNN outperformed the other models to predict the closing price of MANA metaverse cryptocurrency as the four values of metrics record the lowest value. MAPE of the CNN has recorded as 50.8306 means the average of the absolute percentage errors of forecasts is recorded as 50.8306%. The other models, which are LSTM and GRUs, are recorded as means absolute percentage errors of forecasts of 66.6440% and 93.4819%.

**6.5 Experiment 4 Results and Analysis**

The experiment 4 results have been tabulated in table form in metrics of MSE, MAE, RMSE, and MAPE and discussed in this section implicitly. Experiment 4 uses 1 dependent feature as input features to predict the next ten days of closing price, which are labelled as 'Close' target features. 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 individually feeding each feature into LSTM, GRU, and CNN models.

**6.5.1 LSTM**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Metrics | | | |
| MSE | MAE | RMSE | MAPE |
| Open | 4.4785 | 2.037 | 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 | 82.927 |
| 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 |
| Open | 4.4785 | 2.037 | 2.1162 | 72.2716 |
| High | 0.9602 | 0.8349 | 0.9799 | 29.2304 |

**6.5.2 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 | 1.451 | 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 | 3.939 | 3.9574 | 139.3973 |
| SMA\_7 | 13.7488 | 3.6976 | 3.7079 | 130.5644 |
| rsi\_7 | 3.7869 | 1.9374 | 1.946 | 68.3178 |
| EMA\_7 | 19.0914 | 4.358 | 4.3694 | 153.843 |
| Open | 0.8907 | 0.8147 | 0.9438 | 29.0328 |
| High | 6.3392 | 2.5043 | 2.5178 | 88.4685 |

**6.5.3 CNN**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Metrics | | | |
| MSE | MAE | RMSE | MAPE |
| Open | 0.9802 | 0.762 | 0.99 | 27.0086 |
| High | 2.4348 | 1.327 | 1.5604 | 46.468 |
| Low | 2.075 | 1.1681 | 1.4405 | 41.6225 |
| Volume | 0.2294 | 0.4138 | 0.479 | 14.5994 |
| Pre\_Close | 1.3917 | 0.9503 | 1.1797 | 33.6532 |
| H-L | 1.2134 | 0.9027 | 1.1015 | 32.023 |
| 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 | 1.64 | 57.637 |
| EMA\_7 | 1.2483 | 1.0543 | 1.1173 | 37.053 |
| Open | 0.9802 | 0.762 | 0.99 | 27.0086 |
| High | 2.4348 | 1.327 | 1.5604 | 46.468 |

**6.6 Experiment 5 Results and Analysis**

XXXX

**6.6.1 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 |

**6.6.2 GRUs**

**6.XX Conclusion**

With all the three experiments is done in the section above, for the SAND cryptocurrency, the GRU models with the input feature of 'Open', 'High', 'Low', and 'Volume', which are done by experiments 2, perform the best among the other models and input feature to predict the SAND datasets last 10 days price by MAPE of 25.2571. For the SLP, the CNN models with the input feature of 'Pre\_Close', which are done by experiments 1, perform the best among the other models and input features to predict the SAND dataset's last 10 days price by MAPE of 167.2377. In SLP cryptocurrency, all models underperform in experiments 1,2 and 3. With a higher MAPE than 100, the predicted value that differs from the predicted value is very high. For the MANA, the LSTM model with the input feature of 'Open', 'High', 'Low', and 'Volume', which are done by experiments 2, performs the best among the other models and input features to predict the SAND datasets last 10 days price by MAPE of 17.5064.

**CHAPTER 7**

**CONCLUSION**

In this paper, we have introduced LSTM, GRUs and CNN models to make the metaverse cryptocurrency prediction. We have also curated and pre-processed the existing dataset to enrich and increase the dependent feature of the original datasets. The different dependent features are used as the input to make the ten-day prediction closing price of SAND, SLP and MANA metaverse cryptocurrencies based on the last ten days' data. We have achieved the MAPE of 25.2571 and 17.5064 by using the feature of 'Open', 'High', 'Low', and 'Volume' to predict the closing price of SAND and MANA using GRUs and LSTM model. For the SLP cryptocurrency, the best preliminary results we achieved are using 'Pre\_Close' as an input feature, and the MAPE is recorded as 167.2377 by using CNN model. The table of objectives versus progress of this report FYP1 is shown below:

**Table 14: Objectives Versus Progress FYP 1**

|  |  |
| --- | --- |
| Objective | Progression |
| 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. | 30% |
| To design and implement the Convolutional neural networks (CNNs), Long short-term memory (LSTM), and Gated recurrent units (GRUs) machine learning algorithms in the predictive models to forecast metaverse-based cryptocurrencies closing prices. | 30% |
| 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). | 30% |

There are some exciting directions for further study. At first, we can use the different combinations of the input feature inside the curated dataset to make the prediction. By this, we can make a more comparison about which features are vital to the result. Furthermore, hyperparameter tuning can be done by changing the parameter and setting such as epoch, batch size and kernel size to decrease the train and validation loss when fitting to the models. Moreover, changing the architecture of the LSTM, GRUs and CNN models can also be considered a good idea. We can try to finetune them to have better results in the testing data.

**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. Advances 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 advanced 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 Advanced 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 Advances 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 Advances 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 Advanced 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

**APPENDIX 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. | --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. | --Thank you for the comments. |

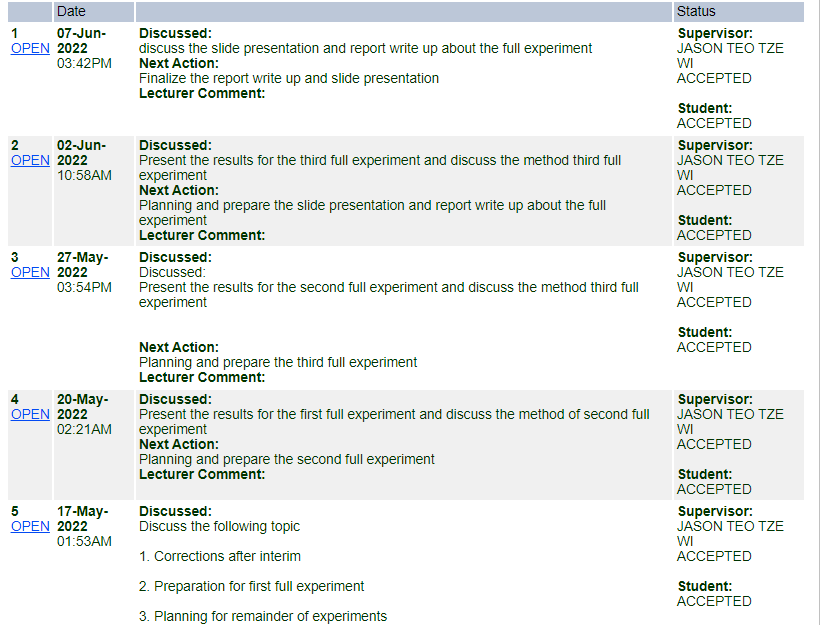
**REVIEWER 3/EXAMINER 3**

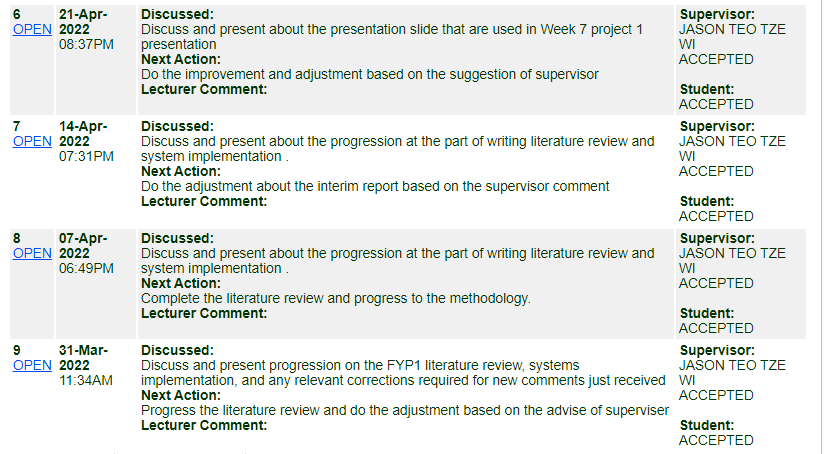
|  |  |
| --- | --- |
| **Comments** | **Reply to Comments** |
| 1. Excellent proposal. The project seems very interesting. Overall the proposal is Ok. However, the literature can be further supported by more recent authentic work in the field. The methodology can also be improved. Some novel deep learning algorithms and latest APIs (i.e., ensemble and hybrid learning models) can help achieve great results. | -The more literature review works was included.  -For methodology ,the LSTM ,CNN and GRUs was used.The hibrid-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.Need to find another new objective. 2. The project scope need to redefine. 3. The problem statements need to improve to show the metaverse cryptocurrencies is no same as the cryptocurrencies. | -The first objective was found and stated in the report.  -The project scope was redefined in the report.  -The citation was included inside the report about the different of volatility of the metaverse cryptocurrencies and traditional cryptocurrencies. |

**APPENDIX B :MEETING LOG SCREENSHOT**





**APPENDIX C :TURNITIN REPORT**