Deep Learning-Based Cryptocurrency Price Prediction in Relation to Trading Volume

Ilia Gershenzon

BSc. (Hons.) Computer Science

March, 2023

I certify that the material contained in this dissertation is my own work and does not contain unreferenced or unacknowledged material. I also warrant that the above statement applies to the implementation of the project and all associated documentation. Regarding the electronically submitted version of this submitted work, I consent to this being stored electronically and copied for assessment purposes, including the School's use of plagiarism detection systems in order to check the integrity of assessed work.

I agree to my dissertation being placed in the public domain, with my name explicitly included as the author of the work.

Date: March 24th, 2023

Signed:

Abstract

The project reported in this paper will aim to investigate the effect of trading volume data on the performance of deep learning models used for cryptocurrency market price prediction.

Among many other crypto coins, Ethereum (ETH) is the coin of choice in this paper, as it will allow a more flexible interpretation of results and comparison to other studies in the field.

Forecasting is made based on the closing price of ETH, and the aim is to predict the closing price for the following week.

The main focus is made on deep-learning-based hybrid models made of Gated Recurrent Units (GRU) and Long Short Term Memory (LSTM), in addition to a customised Transformer model. The performance of the above models was evaluated before and after applying the Volume of day trading as an additional parameter.

The study results show that Hybrid Models do not experience significant improvements after adding Volume. However, Transformer model performance was notably harmed by Volume. Overall, it is not suggested to use the Volume of day trading as an additional parameter when performing cryptocurrency price prediction using deep neural networks.

Table of Contents

1. Introduction

Cryptocurrency is a relatively new digital currency based on blockchain technology and cryptography. It has a decentralised and secure structure. Transactions made in cryptocurrency are nearly anonymous and provide freedom, as it allows users to make transactions without the need for a middleman, similar to banks for fiat currencies. All the transactions are stored in a publicly available blockchain database, which combines transparency and security [1].

The cryptocurrency market as a whole accomplished significant growth in the past decade. The market went from a single Bitcoin to thousands of currencies and tokens [2]. According to CoinMarketCap, there are over 21,000 cryptocurrencies, with a total market value of over $942 billion as of October 2022 [3]. The rapid growth made it very attractive for investors from all over the world.

Cryptocurrency popularity and volatility show the importance of prediction of the price. It is beneficial not only for individual investors as guidance for decision making, but also for financial researchers and studies that will be conducted in the future in the field of market behaviour. Price prediction for cryptocurrencies can be broken down as a popular time-series problem, similar to price prediction for stocks or anything with historical data available [4].

Cryptocurrencies do demonstrate non-linear patterns in price behaviour. When traditional machine learning tools are applied to cryptocurrencies, they perform imperfectly. Hence, there is a need for a more capable prediction tool, such as Deep Learning algorithms. It is a well-known solution for problems that involve forecasting complicated time-series problems [5].

This study will investigate the impact of the volume of trading on the predictive models, as only a few academic papers have considered this parameter. Furthermore, to our best knowledge, nobody measured the effect of trading volume on the hybrid deep learning predictive models, as well as nobody applied a transformer deep learning model for time series forecasting in the form of cryptocurrency price prediction. This will be carried out by creation and testing of hybrid deep learning models and application of a new parameter to evaluate the change in predictive ability. In addition to creation and testing of a transformer model, for the same purposes. The evaluation metrics such as Mean Squared Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) will be used to compare the predictive power of the models before and after the application of volume, and it will show the efficiency of the proposed idea. The information gained from the evaluation will allow for comparisons with previous studies of the cryptocurrency market price prediction.

1.1 Project Aims

The project aims to investigate the impact of the daily trading volume as an additional input parameter on hybrid deep neural networks used for cryptocurrency price prediction. As well as implement the first ever transformer suitable for time series forecasting, apply it to cryptocurrency price prediction and test the impact of volume. Discuss the change in the predictive power of the models and make a conclusion. In order to collect the data suitable for the investigation, the following objectives will be utilised:

- Create LSTM-GRU and GRU-LSTM hybrid models, train using ETH as the coin of choice and measure performance – to get the baseline data for the study.
- Create a Transformer model suitable for time series prediction, train using ETH as the coin of choice, and measure the performance – to get the baseline data for the study.
- Discover the optimal input window size for the above models.
- Discover the optimal optimisation formulas for the above models.
- Apply Volume as an additional parameter for presented models, experiment with different normalisation formulas, and measure models' performance after the input parameter changes.
- Make a conclusion based on collected data and present a verdict on the effect of volume as an additional parameter on hybrid deep learning models and Transformer.

1.2 Report Overview

Background - This section will describe research projects previously conducted in cryptocurrency price prediction, as well as methods and principles that were used before. Focus is made on hybrid deep learning models, as it is one of the targets of the project. Along with an overview of work done in the field of transformer models, the logic behind them and studies utilising these principles.

Data Preprocessing - The chapter outlines the choice of coin for the project, the data sources used, and the operation performed with raw data before any application in the models.

Design - In this section, the focus is made on the specific design ideas and challenges that were faced in the progress of the project's creation.

Implementation - This chapter focuses on specific details of the implementation part of the project. The section describes hybrid deep learning models and transformer model that were designed and created. The section further discusses the experiments conducted to determine the best input window for the models and the normalisation techniques applied to input data.

Results analysis - Pre-ultimate section discusses the findings and results of the study. This chapter presents the final prediction results of the models in graphs and evaluation metrics.

Conclusion - It is the final chapter of this report, with a discussion about the completed project, possible further work and lessons learned.

2. Background

This section provides an overview of work previously done in the cryptocurrency price prediction field and a description of research papers that were used as the theoretical basis for the transformer model implemented within this project.

2.1. Hybrid Deep Learning Models

Previous projects relating to this study include deep learning-based cryptocurrency price prediction schemes with inter dependent relations [6]. The paper reports on how the use of parent coin data affects the hybrid predictive models. As the parent coin, Bitcoin (BTC) is used because it is the oldest and most valuable coin on the market. Coins for prediction were Litecoin and Zcash, both are relatively old and historically replicate the movement of BTC market price. Public data for training and testing the models was taken from investing.com [7], a trusted information source in the crypto community. For the prediction model, two types of neural networks were used: Long short-term memory (LSTM) and Gated recurrent units (GRU); flattening was also used on the parent coin's data. As mentioned previously, models were combined into a hybrid model, the output of which was combined with a flattening result, producing a realistic and accurate prediction for windows of one, three, seven and thirty days.

However, this paper focused on the average price parameter. It used it as the primary information for hybrid deep learning neural networks, so conclusively, it only shows the predictive ability with this limitation. Therefore, applying more types of publicly available data: open price, high of the day, low of the day, and the daily volume of trading; could show how the predictive ability of the model changes depending on the input data. Such changes in predictive abilities in deep neural networks will be studied in this project. Moreover, only mean squared error (MSE) was used to evaluate the proposed model, making it more complicated to compare to other papers in the field. Therefore, applying other evaluation metrics could lead to better comparable results.

Furthermore, the Ether price prediction techniques used by Politis et al. [8] also relate to our project. The project involved price prediction for Ethereum (ETH), or Ether, as it is referred to in the study. The idea of a parent coin was also reflected in this study, and the price of Bitcoin was used to reflect market dynamics. Also, the price of ETH and volume in USD were used, as well as some technical indicators: simple moving average (SMA) of two weeks, exponential moving average (EMA) for the same two weeks and Moving Average Convergence Divergence (MACD). Additionally, popularity indexes from Google trends for terms (ETH, Coinbase and Exodus) were used. However, two attributes of the ETH blockchain were used as main characteristics: daily block size and mining difficulty. For prediction in this study were used LSTM, GRU and Temporal Convolutional Network (TCN) models; also, the different combinations of them in the form of hybrid networks were tested:

LSTM-GRU, LSTM-TCN, GRU-TCN. A comparison of the results for all six models made this study comprehensive. Furthermore, root-mean-square error (RMSE) and mean absolute percentage error (MAPE) were used to evaluate the results, which made the work easily comparable with other works in the field. Generally, hybrid models outperformed individual ones, as concluded by the study.

The above study showed how accurately the price of ETH can be predicted using different hybrid deep-learning neural networks. However, data used as main characteristics is no longer relevant to the Ethereum blockchain. A few months ago, ETH switched from the Proof of Work (POW) protocol to Proof of Stake (POS). In the case of POS, the block in the blockchain does not have difficulty anymore. The number of blocks is fixed per unit of time. Hence, one of the main parameters is not usable anymore; it could be interesting to test the same network structures with a different and more relevant set of main characteristics. This project will also involve the usage of hybrid neural networks but comparing it with other hybrids, not with individual models. Moreover, it will focus on not out of date characteristics of blockchains.

Another paper investigated the price prediction capabilities of deep neural networks in the case of cryptocurrencies [9]. As an object for the experiment, Bitcoin was chosen as a market maker in the world of cryptocurrencies. Prediction of the BTC price will give the overall view of the cryptocurrency market. LSTM and GRU models were chosen and tested, and the goal was to determine the best accuracy and compilation time. GRU outperformed LSTM in compilation time and demonstrated a better ability to predict time series-related problems. Daily price values were used as a base of the dataset, including opening price, high price, low price, closing prices, and market capitalization for the day. The data was collected in the period from 2014 to 2018.

However, many daily price values were used in a data set. It was still not including the volume of trading. Also, data from only four years was utilised, which does not include the last four years of significant changes in the market. Furthermore, the project focused on LSTM and GRU models in non hybrid format, which makes the purpose of the study doubtful as it is proven that singular models' performance in such tasks is lower than hybrid ones. Therefore, it would be practical to repeat the project with an updated data set and more parameters in consideration, as well as hybrid models under evaluation.

As a result, our project will specifically investigate hybrid neural networks for cryptocurrency price prediction. Hence, they are the most efficient ones, and only a few experimentations with input parameters have been made, which makes it a challenging but exciting task to reinforce existing network models with volume data and discuss the results.

2.2. Transformer Models

Transformer models were proposed in 2017 by Vaswani et al. in the paper "Attention Is All You Need" [10]. The model proposed in the paper was an innovation in the world of dominant sequence transduction models; such models usually use complex recurrent or convolutional neural networks with an encoder and decoder. The complexity of the above models makes them expensive to train and test as they require a lot of computation power. On the other hand, proposed Transformer networks use only attention mechanisms and avoid computationally expensive recurrence and convolutions. Instead, they use auto-regression, which allows newly generated symbols to rely on previously generated symbols.

Transformers from the paper above were used for two machine translation tasks. One model improved the best results for the WMT 2014 English- to-German translation task. While the other outperformed all the competitors in WMT 2014 English-to-French translation task. Although training time for transformers was a lot less than for any other state-of-the-art models, both attention models achieved impressive results. Additionally, new models were applied to English constituency parsing tasks and demonstrated promising results.

Transformers or attention-based neural networks demonstrated promising results in text-related tasks, such as translation and constituency parsing. However, they were not applied to anything but text-related work, and taking into account their outstanding performance; it is the perfect next step to try and apply innovative Transformers to other jobs, for example, time-series prediction and see if they are capable of competing with state-of-the-art solutions like hybrid deep neural networks.

After the Transformer models were proposed, they gained much attention from the scientific community. Furthermore, one of the most popular applications of Transformers models for natural language processing (NLP) was found by Brown et al. [11] and proposed in the form of a 175 billion parameter autoregressive language model, also known as GPT-3. Moreover, GPT-3 is a prominent ancestor of the very famous chatGPT, as ChatGPT is essentially a supervised fine-tuning version of GPT-3.

GPT-3 Transformer demonstrated excellent performance in many NLP tasks. The model was tested in zero-shot, one-shot and few-shot, where the number of shots refers to a number of example task solutions given to the network before evaluation. Generally, the goal was to test Transformers' ability for rapid adaptation to different tasks. Broadly, GPT-3 achieved good results in zero-shot and one-shot settings and even outperformed state-of-the-art models in the few-shot setting. Of course, there were tasks with low performance, including language inference tasks and comprehension datasets reading.

Overall, GPT-3 demonstrated impressive results in NLP tasks even without fine-tuning and limitations in the number of examples. However, all the tasks were related to NLP or other language processing datasets and tasks. So still, no study applies such innovative and robust network architecture to any other area of DNN problems. In our study, we are applying the Transformer model for time-series prediction, or cryptocurrency price prediction, to be precise.

Now it is time to discover what work was done in the field of cryptocurrency price prediction in relation to Transformer models. There are only a few studies that contributed to that area;

one of them is [12]. In the paper, transformer models of two types are used to analyse the social media comments, in particular Tweets made on the topic of Bitcoin. The study focuses on the prediction of Bitcoin price using Close value and tweets related and split into three categories: positive, negative and natural. FinBERT and RoBERTa models were trained and used for the tweeter's feed split. Later the LSTM, GRU and vanilla RNN were used to convert the labels made by transformers and closing price into the actual prediction. However, the study concludes that the model at the end of the day was not accurate enough in the given state. The paper concluded that it is not possible to create an accurate enough prediction model for real world use. However, by working on the accuracy of the training data set and by increasing the size of the training data set combined with hyperparameter tuning, the predictive power of the model could become sufficient.

The paper above uses transformer models but in a very different way than we are going to use them. In our study, we are trying to substitute DNN with a transformer and place it in a new and unusual environment to evaluate its performance on an entirely new task. Moreover, in [12], transformers were used only in a new type of text analysis task, and it is what transformers were created to do initially. There is nothing new in such a way of using them. That highlights the difference between the ideas used in this paper and the above one.

Another study in the field of cryptocurrency price prediction is [13]. The Transformer was used in combination with LSTM; precisely, the results of the two models were combined using a hybrid combiner, so the models were used in parallel. Tanwar and Kumar [13] used a very different type of transformer model. However, it utilised the same shared principles with our implementation: self-attention and positional encoding. Overall, it was used in hybrid with LSTM to predict prices of Bitcoin, Binance coin and Ethereum. The experiment showed that the Transformer and LSTM model took the longest time to compile but was the most accurate when compared to three different DNNs: LightGBA, Choudhary Cnn and LSTM-GRNN. The conclusion was made that LSTM, along with Transformer, give the best performance for time series prediction out of the three tested models.

For the above reasons, the authors of the paper Prediction of Cryptocurrency prices using Transformers and Long Short term Neural Networks [13] did a great job using the transformer model for cryptocurrency price prediction. However, there are fundamental differences in to work that is performed in our study. The transformer model performance is not evaluated without any hybrid components, and the model uses bagging, which is very computationally expensive and makes the model very time and energy consuming to compile and run. The previously mentioned reasons indicate the fundamental difference between our subject of study and all the papers published before.

Overall, transformers have proven to be very competitive in language processing tasks and were utilised in combination with other RNNs for cryptocurrency price prediction. However, there were no cases when price forecasting was performed entirely by the appropriately constructed transformer model alone. That is the area where our work will try to fill in the gap and make the conclusion regarding the question of can transformers be used for time-series forecasting in the setting of the super volatile cryptocurrency market.

3. Data Preprocessing

This chapter discusses the choices made regarding the cryptocurrency data used in the project for models training and evaluation. Additionally, data sources are explained, and operations on raw data are described.

3.1 Dataset Description

Ethereum was chosen because of several factors. Firstly, while Bitcoin (BTC) is a market maker of cryptocurrencies, it reflects the overall dynamics of all cryptocurrencies and is frequently used as additional parameters for DNN or applied as a parent coin [6, 8]. To make it easier for peers to review this paper, interpret the results and compare it to papers that use BTC, the other major coin in the crypto market was considered - ETH. Secondly, Ethereum is a very old cryptocurrency that had a massive impact on the market as a whole. It would be very beneficial for investors to know the future price fluctuation of ETH.

Data for the project was taken from a well known source of financial data: Yahoo finance [14]. It is a frequently used source of information for financial studies, stock prediction studies and time-series research papers [15, 16]. It is a trusted source of information not only for stock market data but also for the cryptocurrency market. Because of these reasons, Yahoo finance is a chosen source for ETH trading figures.

When downloaded from Yahoo finance, the ETH data table has the following fields: Date, Open, High, Low, Close, Adj Close, and Volume. The values follow the daily format. Most of the studies in the field use daily data, as reported in the cryptocurrency price prediction survey [5]. Only Date, Close, and Volume will be used in our implementation. Our models will be predicting the ETH closing price for each day of the following week.

A sample image of the used dataset is shown in Figure 3.1. It demonstrates the fields and values of the utilised data. Our dataset consists of seven columns and 1828 rows. We are showing the first few rows of data in the image below.

	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Adj Close	Volume
2	2017-11-19	347.401001	371.290985	344.739990	354.385986	354.385986	1181529984
3	2017-11-20	354.093994	372.136993	353.289001	366.730011	366.730011	807027008
4	2017-11-21	367.442993	372.470001	350.692993	360.401001	360.401001	949912000
5	2017-11-22	360.312012	381.420013	360.147003	380.652008	380.652008	800819008
6	2017-11-23	381.438995	425.548004	376.088013	410.165985	410.165985	1845680000
7	2017-11-24	412.501007	480.972992	402.757996	474.911011	474.911011	2292829952

Figure 3.1: ETH price dataset

In the table above the columns represent the following:

- Date - time of the record in format YYYY-MM-DD
- Open - price at the start of the day
- High - highest price of the day
- Low - lowest price of the day
- Close - price at the end of the day
- Adj Close - is a metric used mainly in the stock market and it represents Close price adjusted according to all dividend distributions and splits that accrued during the day. However, it is not applicable to cryptocurrencies, so the Adj Close in our case is equivalent to Close.
- Volume - represents the total volume of trading in USD that happened in a day

All the format changes that took place before model training will be described in the Dataset Preprocessing section of the report.

3.2 Dataset Preprocessing

Data preprocessing included two steps. Firstly, conversion of the Date field in the downloaded table to ‘datetime64’ format instead of ‘object’ type. It is a crucial step because the Date field in a different format could break functions that are further used to draw graphs with predictions of models. The second step was to convert Volume from ‘int64’ to the type of ‘float64’, similar to other fields in the table. It is important to have Close price and Volume of day trading in the same format, as they later will be used together, and type differences can affect normalisation procedures. The modified file is saved back to the directory of origin.

After preprocessing steps, the data is presented as a ‘.csv’ file with Date, Open, High, Low, Close, Adj Close, and Volume in appropriate format types and ready to be used in the models' implementation.

4. Design

This section of the report discusses major design challenges that were faced during the development of the models and project progression. Understanding of these design decisions will be important to the overall flow of the implementation part of this project. This section will be focusing on general aspects and problems that will be further discussed and supported with evidence from literature or experiments in the implementation section.

The fundamental parts of the project are Hybrid models and Transformer model. These are completely different DNNs and both cases required separate design approaches and decisions to be made. They utilise different principals and logical components. Hence, this section is split into two subtopics one discussing the design of Hybrid models and the other about Transformer.

Another important stage of initial design is shoes made regarding the implementation environment and programming languages to be used within this project. All the bits of the models were created using Google Colab. It allowed free usage of sufficient computation power, it had all the Python modules needed preinstalled and saved a lot of time usually spent on solving problems with the operating system and modules incomparability. In addition to it, Jupyter Notebook formation of code that is handy for data science related projects and allows code to be clear and structure for any reader due to extensive commenting and segmentation. Regarding language and libraries used more information is provided in the beginning of implementation part of this report. Generally, the Tensor Flow and PyTorch were used for models implementation. Tenser Flow was used for Hybrid models and PyTorch for Transformer.

4.1 Hybrid Models

There are many models that can be used for time-series tasks. Hence, firstly, it is important to say why the hybrid deep learning models (HDL) were used in the project rather than simpler non-hybrid versions or machine learning algorithms. It was proven by Pintelas in 2020 [4] that deep learning models (DL) outperform machine learning (ML) in time-series problem-solving when applied to cryptocurrencies. At the same time, HDL models beat DL models in cryptocurrency price prediction, as was concluded by Politis [8]. We want our research to utilise the most efficient models as they will be used a lot in the future, which is why we implemented and conducted experiments on HDL.

The other question when constructing HDL models is what artificial neural network components to use. In order to answer, we have to look back on the previous research papers in the field. Recurrent neural networks, or RNNs, feature the recurrent connection between the input and output layers and can follow contextual information along with data input. That makes them great for long sequences of data. Two common RNN networks are LSTM and GRU [9]. The ability of LSTM and GRU to effectively manage past information makes these architectures state-of-the-art solutions for problems involving sequential data [8]. Additionally, separately or combined above RNNs were used in many studies across the field of cryptocurrency price prediction [4, 6, 8, 9]. Based on the previously stated reasons, it was apparent to choose LSTM-GRU and GRU-LSTM hybrids for this project to make it as relevant as possible to the current and future studies.

In the proposal of that project, it was suggested that we would be able to find an academic paper with available source code to utilise for our HDL models implementation. However, after spending time searching and sending emails to authors of papers on cryptocurrency price prediction, it became clear that there is no one who would want to share the source code or anybody who initially posted it to open source.

Finding a paper with accessible source code is highly complex for HDLs in cryptocurrency price prediction. Therefore, the design for our models was partially taken from the study about stock price prediction [15]. They were not using hybrid models; however, our hybrid models follow the same structure regarding the number of dense layers and output and input parameters. Moreover, the results of models are comparable, especially training and validation loss graphs that show if the model is learning correctly. However, the accuracy of models is hardly comparable as the cryptocurrency price is way more random than the stock price [6], so it is harder to predict. Hence, the accuracy of our models is lower.

When the structure of the models is clear, the next step is to determine the input and output parameters and windows. The input and output window represents the number of data points to be fed into the model and produced as the prediction result.

The size of the input window is subject to change from study to study, and there is clearly no uniform solution for all the problems. The exact answer needed to be revised in any previously conducted study; there was no conclusion, so the decision was made to conduct an experiment that would allow us to determine the best input window for HDL models in our particular case. Hence, our data points and our models could be very different from the existing ones, and they could suffer performance issues if all the subjects to change parameters are not carefully picked. The experiment was conducted in the progress of models implementation, and the results of it are present in the implementation section of the report. Similarly to the input window, the size of the output window is justified in the implementation part as well.

Normalisation is another important technique needed for the creation of an accurate deep learning model. There are many normalisation methods that can be applied to our input data. However, choosing good normalisation that is suitable for data could dramatically improve the performance of HDL models. In our case, we are considering two models: LSTM-GRU and GRU-LSTM, and each of these models can react individually to different normalisation formulas. To ensure the best possible method found for our models, we decided to do another experiment that included two models and a number of normalisation techniques. It will allow us to clearly see which formulas best satisfy the ETH price data points and needs of HDL models.

When the model's format, input and output windows and normalisation methods were known, it was time to go on to design the new set of models that would be able to take the daily Volume of ETH trading into account. These models are going to need a different input window as the input data consists of two different metrics. Additionally, the normalisation could affect the new data set differently, so the experiment with different normalisation formulas will be repeated in the setting of the new input parameter.

Later the performance results of both model sets will be compared, and the conclusion will be made on the topic question of this research.

4.2 Transformer

The transformer model (TM) is a short name for the model implemented in this study because, in reality, it is a decoder only transformer that combines a number of techniques proposed in different studies throughout the past seven years.

The original transformer model (OTM) proposed in Attention Is All You Need [10] has an encoder and decoder. The model is made to perform translation operations. In translation, there is an input and output, and they communicate with each other to produce the most accurate translation. In the translation task, there is no separation into future and past events. However, in time-series forecasting, the input cannot talk to the output as the output has not happened yet; it is in the future, while the input is in the past. For our purposes, it is vital to ensure that future tokens cannot communicate with the past. Hence, our transformer uses only a decoder.

TM generally uses a lot of techniques proposed in [10]. In more detail, all that methods will be addressed in the implementation part of the report, especially in the Transformer chapter. Below, here we are going to briefly list the components of the Transformer model.
Our transformer will include the following parts:

- Triangular mask - will be used to ensure that there is no communication between future data and past data.
- Scaled Dot-Product Attention - this concept lies in the heart of any transformer model; it is the most important and innovative part that allows tokens to communicate with each other.
- Positional encoding - is essential for attention to work correctly; it makes sure that data points know where they are in regard to future, past and present, or in simpler words, where they are positioned in the input sequence.
- Multi-Head Attention - this principle speeds up the computation of the model by splitting convolution into many groups.
- Feed-Forward Network - helps tokens to ‘think’ about their neighbours.

- Residual blocks - are made to avoid the vanishing/exploding gradients problems.
- Layer Normalization - reduces training time and normalising the neurons.
- Dropout - was made in case the model would have an overfitting problem.

After the implementation of all the above parts of the TM, it is time to solve the last problem of Transformer - input. Usually, transformers work with some form of the alphabet (letters or subword blocks), and input is encoded as a string of values representing the position of the symbol in the array or alphabet. In our case, the “alphabet” will be constructed out of unique data points, and the input will be encoded in a similar way to the original manner.

When the TM is created and ready to be used, we will also study the effect of different input windows and normalisation methods. It is going to ensure good results that are easily comparable with hybrid models. After it, the last step will be to change the TM according to the new input parameters, which will also include the Volume of day trading for ETH.

The next chapter will focus on the implementation details for all the models previously described.

5. Implementation

In this chapter, the implementation process will be discussed. It is split into two parts, one discussing the creation of hybrid deep learning models and the other creation of a transformer. For hybrid models, it will include the general information behind the implementation process and information about input window experiments and experiment with different kinds of normalisations applied to volume data later on.
The hybrid models part will be split into two subtopics:

- Implementation before volume - how the initial models were implemented, what issues were addressed in the process, and how they were resolved.
- Implementation with volume - how the models were changed to add a new parameter in the form of the day trading volume of ETH.

The transformer model section will include an explanation of methods utilised in implementation and the theory behind those methods, on top of a general explanation of what the transformer models are and what type of the model is implemented in this project. Similar to the hybrid models section, this segment will include the part with general and volume-related implementation.

Regarding the languages and libraries used in the implementation: Python and TensorFlow were used for Hybrid Models, and the same library and language were used in inspiring work [15]. The transformer part utilised Python and PyTorch because PyTorch allows customising the network parameters and layers; this feature was essential to create the transformer.

5.1 Hybrid Models

5.1.1 Before Volume

The reasons why HDL or hybrid deep learning models were used are stated in the design section of the report above. In this part, the focus is made on the implementation decision and experiments that were previously described. The initial implementation of the model was not a complex task that included following the structure present in [15] and adding a proper second layer (GRU or LSTM). When models were created and tested, it was time to start the experimentations needed to create accurate and trustworthy models.

The next value to consider in the implementation of models was the output and input windows size. Output window could be any size from one day to as many as we want. One day is the smallest output possible because our data is daily. The decision was made in favour of a 7 days output window because it is one week of trading in the cryptocurrency market. Similarly, in the study [15], we are replicating models from a five-day output window that was used to represent one week on the stock market.

Regarding the input window, the experiment was conducted with the implementation of models with four different input sizes to see how the input window affects the models' performance. The experiment included a one, two, three and four weeks window. The results showed that more days in input for the HDL model produces less accurate results. The most accurate model was achieved for LSTM-GRU and GRU-LSTM with one week input format (see Table 5.1).

	Input In Number Of Weeks			
	1	2	3	4
LSTM-GRU	MSE: **31126**	MSE: 86199	MSE: 115723	MSE: 138802
	MAE: **130.29**	MAE: 238.89	MAE: 306.89	MAE: 332.77
	RMSE: **176.42**	RMSE: 293.59	RMSE: 340.18	RMSE: 372.56
	MAPE: **9.41**	MAPE: 17.81	MAPE: 22.89	MAPE: 22.09
GRU-LSTM	MSE: **38166**	MSE: 42942	MSE: 55204	MSE: 151917
	MAE: **141.44**	MAE: 161.26	MAE: 195.84	MAE: 307.31
	RMSE: **195.36**	RMSE: 207.23	RMSE: 234.96	RMSE: 389.77
	MAPE: **10.22**	MAPE: 11.77	MAPE: 13.76	MAPE: 22.49

Table 5.1: Performance of LSTM-GRU and GRU-LSTM models. Best results in **bold**.

The final structure of both models can be seen in the figure below (see Figure 5.1). These models will be the baseline of our study, and any further comparison will utilise their performance metrics.

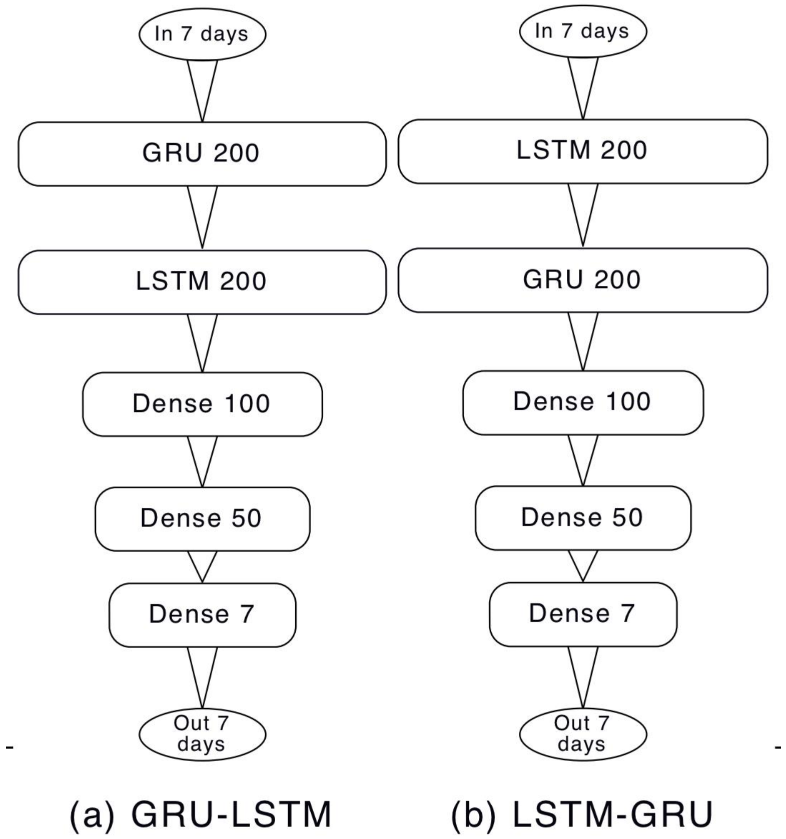


Figure 5.1: Structure of HDL. Left: GRU-LSTM model. Right: LSTM-GRU model.

The next step is choosing the best normalisation technique for our task. The application of normalisation methods to baseline models will allow us to see the performance difference. The effectiveness of any DNN depends a lot on the normalisation method. As per suggestions in [17, 18], we will apply more than one normalisation. For our experiment, the three most common methods were taken from the studies evaluating data normalisation for DNN, especially in time-series forecasting and stock index forecasting [17,18]. The three methods are Z-score, Min-Max and Log scaling normalisations. Initially, the list was different, but the experiment showed that only a few methods could cooperate with high Volume values in the data set. For example, the very promising Tanh estimator and Decimal Scaling performed exceptionally poorly, and performance evaluation values for both were on the level of RMSE = 0.567 and MAPE = 760.6. For comparison, the Log scaling chosen by the experiment showed: RMSE = 0.146 and MAPE = 1.46 (reminder: lower RMSE and MAPE values are better). Overall, after several experiments, Z-score, Min-Max, and Log scaling showed the best results and were chosen as our study's final set of methods.

Applying three kinds of normalisations to proposed models showed a significant performance increase, as suggested in [17,18]. The table with the evaluation is presented below (see Table 5.2). It is clear from the table that Min-Max and Log scaling show the best possible results and outperforms Z-score normalisation.

	Normalisation Method			Best Results Before Normalisation
	Min-Max	Z-score	Log scaling	
LSTM-GRU	MSE: **0.0018**	MSE: 0.0901	MSE: 0.0214	MSE: 31126
	MAE: **0.0305**	MAE: 0.2485	MAE: 0.1058	MAE: 130.29
	RMSE: **0.0424**	RMSE: 0.3001	RMSE: 0.1463	RMSE: 176.42
	MAPE: 11.28	MAPE: 88.4217	MAPE: **1.47**	MAPE: 9.41
GRU-LSTM	MSE: **0.0019**	MSE: 0.0371	MSE: 0.0253	MSE: 38166
	MAE: **0.0314**	MAE: 0.1419	MAE: 0.1175	MAE: 141.44
	RMSE: **0.0438**	RMSE: 0.1927	RMSE: 0.1589	RMSE: 195.36
	MAPE: 11.71	MAPE: 52.78	MAPE: **1.63**	MAPE: 10.22

Table 5.2: LSTM-GRU and GRU-LSTM performance metrics with 3 types of normalisation and compared to best results obtained before normalisation applied. Best results in **bold**.

The next step was to apply the same normalisation methods to Volume and Closing price and create a new set of hybrid models that uses a new combination of data.

5.1.2 With Volume

Now that the format of the models is known and the best input, output windows and normalisation are justified, it is time to start rebuilding the models to include volume as input paired with closing price. Based on the results from the experiment with the input window, further changes will be made only to 1 week in 1 week out models because they proved to be the best ones.

The volume will be added as a second dimension in the input layer of the models. Hence, the format of the input window will be 7, 2 instead of 7, 1. In this case, we get the situation where for each closing price value, there is one volume of trading value. Both values are normalised separately and then sent to the input layer.

In order to see the effect of normalisation, both models are trained on 3 data sets each, where each data set is the result of one of the normalisation techniques. For each case, the number of essential parameters were optimised, such as learning rate, size of each layer and dropout if needed. The performance results can be found below in table 5.3.

	Normalisation Method		
	Min-Max	Z-score	Log scaling
LSTM-GRU with Volume	MSE: 0.0025	MSE: 0.0637	MSE: 0.0199
	MAE: 0.0376	MAE: 0.1864	MAE: 0.1055
	RMSE: 0.0503	RMSE: 0.2523	RMSE: 0.1413
	MAPE: 14.19	MAPE: 77.6355	MAPE: 1.46
GRU-LSTM with Volume	MSE: 0.00129	MSE: 0.0446	MSE: 0.0228
	MAE: 0.0400	MAE: 0.1493	MAE: 0.1070
	RMSE: 0.0542	RMSE: 0.2111	RMSE: 0.1513
	MAPE: 15.34	MAPE: 67.535	MAPE: 1.48

Table 5.3: LSTM-GRU and GRU-LSTM performance metrics with Volume and 3 types of normalisation.

The table shows that Log scaling outperformed Z-score and Min-Max normalisations. Overall results will be discussed further in the Results section of the report.

5.2 Transformer Model

5.2.1 Before Volume

From the design section, it is clear that TM will be a lot different from the GPT-2 or OTM; it will be missing some components because of the nature of time-series forecasting. However, all the remaining components will be described in great detail with examples, pictures and tables in this section of the report.

Because of the reasons above, our implementation can be represented in the picture below (see Figure 5.2); it also includes the OTM and shows the difference in the design of both.

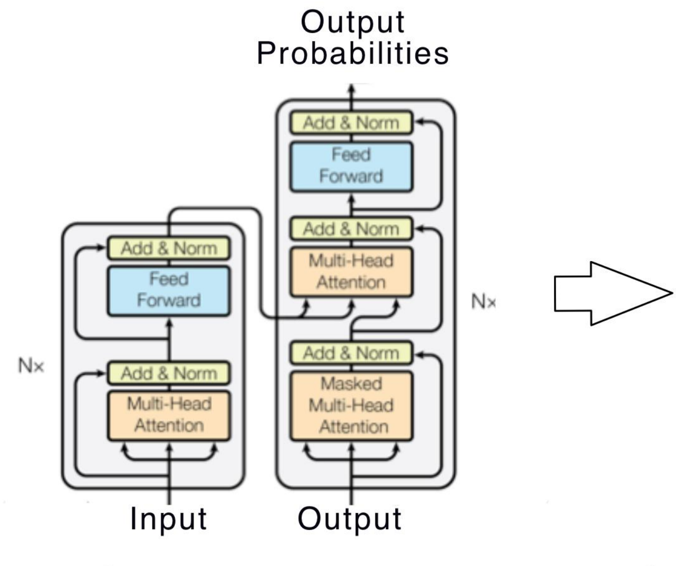


Figure 5.2: OTM and TM graphical representation. Left: OTM Adapted from [10, Fig. 1].
Right: TM model.

In order to ensure the inability of information exchange between future tokens and past tokens, the triangular mask is applied to input data. It makes the future tokens' values equal to 0 while the past have some value. The example below demonstrates the application of a triangular mask on a list of three elements (see Figure 5.3).

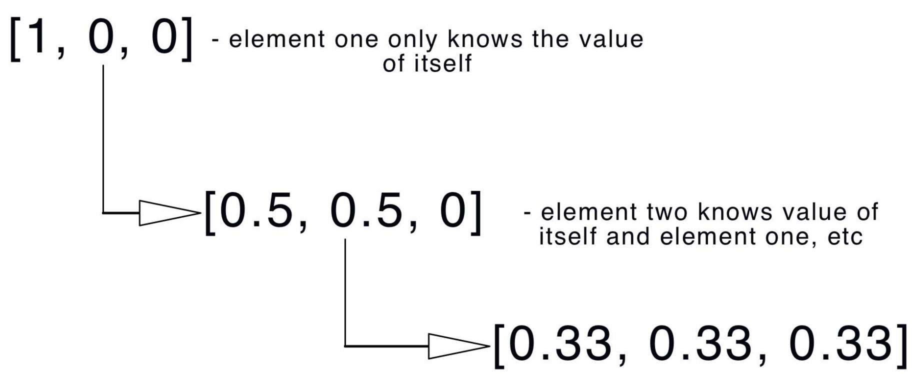


Figure 5.3: Example of triangular mask applied to list of 3 elements.
Note: Values are examples only

The power of the Transformer model lies in the attention; it is the ability of tokens to look back on each other and adjust values accordingly. In TM, we use "Scaled Dot-Product Attention", which was used in OTM and proposed in the same paper [10]. The idea is that each token has a query and key. Where the query is what values taken wants to see behind it, and the key is what value the token assigns itself. Then, the dot product of the query with all keys is computed and divided by the dimension of queries and keys $\sqrt{d_{q\&k}}$. Next, softmax is applied to obtain the weights on values.

The output matrix is computed as in [10, eq. (1)]:

$$Attention(B,\ C,\ V)\ =\ softmax(\frac{BC}{\sqrt{d_{q\&k}}})V \qquad (1)$$

Where B is a set of queries, C is a set of keys and V a set of values.

For attention to work correctly and for tokens to be able to look back on their neighbours, they need to be aware of their position in the sequence. For these purposes, it is crucial to apply positional encoding. It will allow tokens to understand where they are in sequence and who is behind them [10]. It is possible by the addition of a token and position tables. Both are generated using the PyTorch Embedding module, which allows to map index values to a weight matrix and processes discrete input symbols or values in a continuous space. Initially, tables are created with random values between -1 and 1, but during the training phase, the values are updated via backpropagation to minimise the loss function.

When single attention and positional encoding are implemented, it is time to make Multi-Head Attention (MHA) [10], another principle used in OTM. Instead of using one head of attention at a time, in MHA h number of attention heads are performed and calculated in parallel. Afterwards, they are concatenated to create a single vector. Instead of doing one large convolution, MHA makes convolution in smaller groups, allowing better performance [10].

To this point, our TM has the following parts implemented (see Fig. 5.4): Self Attention that is converted to MHA and masked using our triangular mask.

Figure 5.4: Masked Multi-Head Attention

In the original paper, the subsequent implementation step uses the Position-wise Feed-Forward Network [10]. Thanks to MHA and triangular mask, this mechanism allows tokens that already know their neighbours from the past and themselves to analyse new information and make conclusions about their surroundings. Feed-Forward Network in our implementation has a form of one linear layer followed by a non-linearity, and it is called after each call of Self Attention. It applied to each node separately and identically.

The position of the Feed-Forward block on our overall model diagram is shown below (see Figure 5.5).

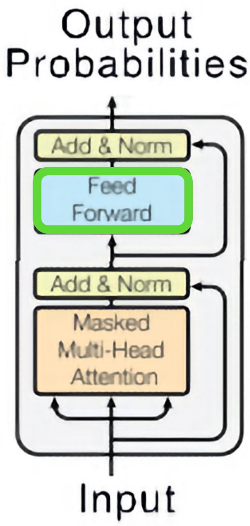


Figure 5.5: Feed Forward block

The only parts that are now missing are a skip connection or Residual blocks and Layer Normalization (see Figure 5.6). Residual blocks were initially introduced by He et al. [19] and used in OTM implementation [10]. The Residual block's purpose is to help deep networks that start to experience vanishing/exploding gradient problems. To overcome these issues shortcut connections are introduced. They allow values to skip one or more layers in the network, and later they are added to the output of the stacked layers [19]. Layer Normalization is applied before adding the skip path to the main branch [10]. It can substantially reduce the training time and make the network more efficient by normalising the activities of the neurons [20]. It was originally proposed in [20] and implemented in PyTorch [21]. We will use a pre-implemented version of Layer Normalization available in PyTorch.

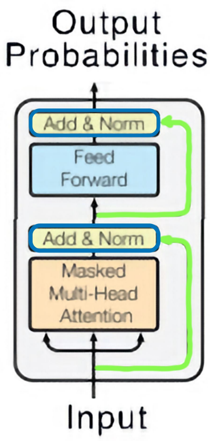


Figure 5.6: Residual blocks in green and Layer Normalization in blue.

Overfitting problems arise when networks reach a high level of complexity [22]. Transformer DNNs usually require a large number of parameters; for example, in the OTM paper, the number of models were used starting from a size of 28×10^6 params and way up to

213×10^6 params in the most significant model [10]. In order to fix this issue and ensure that our model can be safely scaled up for future research purposes, we have implemented a Dropout. Dropout is a technique created to prevent overfitting in neural networks. It turns off some percentage of randomly chosen units in the layer and prevents units from overly relying on one another [22]. In our implementation, dropout can be applied at every stage of the training process, so no hidden layer of TM will suffer from overfitting.

The last step in the TM implementation was to convert the data about the closing price of ETH into a suitable format to feed it to TM. Originally transformers are made for the prediction of the next sub-word block or letter in the output [10, 11]. It means that they learn on the set of values where words are converted to sub-words, sub-words make a known alphabet for the current problem and input encoded into a numeric string using positions in the alphabet list. In our case, the ETH closing prices were converted into an “alphabet” that holds unique values from the data set; in this case, unique values equal “sub-word” blocks. Further, the input string with closing values is encoded using positional representation in the alphabet list. It allowed us to convert values from closing prices into integer numbers limited by the size of the alphabet list and feed them into TM. When the prediction is made, the result is decoded and presented as the ETH closing price. Examples of encoding and decoding with the example alphabet are presented in Figure 5.7.

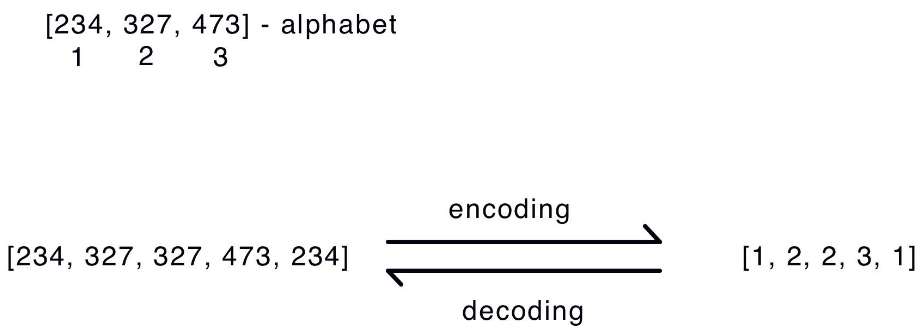


Figure 5.7: Example of Decoding and Encoding

Now when all the building blocks of the final structure are implemented, we can see how they all are represented in Figure 5.2 (on the right) and together construct the Transformer Model suitable for cryptocurrency price prediction.

The input window experiment was conducted on TM. Similarly to HDL models, four types of inputs were tested (See Table 5.4). Results from the table below help to conclude that input window of one week is the best for proposed TM.

	Input In Number Of Weeks			
	1	2	3	4
TM	MSE: **335893**	MSE: 410843	MSE: 508611	MSE: 1184579
	MAE: **347.25**	MAE: 419.65	MAE: 396.5	MAE: 945.78
	RMSE: **579.56**	RMSE: 640.97	RMSE: 713.17	RMSE: 1088.38
	MAPE: **25.33**	MAPE: 28.16	MAPE: 27.43	MAPE: 64.50

Table 5.4: Performance of TM with different input windows. Best results in **bold**.

The last part of TM development now left is to see how previously tested normalisation methods will affect the performance of the TM. In testing, the same normalisations as for HDLs were used. The results are demonstrated in the table below (see Table 5.5). It is clear from the normalisation experiment that TM benefits from normalisation; however, less than HDL models. MAPE improvement presents only when Log scaling is applied.

	Normalisation Method			Best Results Before Normalisation
	Min-Max	Z-score	Log scaling	
TM	MSE: 0.0355	MSE: 0.4974	MSE: 2.6672	MSE: 335893
	MAE: 0.1633	MAE: 0.5746	MAE: 1.3921	MAE: 347.25
	RMSE: 0.1884	RMSE: 0.7053	RMSE: 1.6331	RMSE: 579.56
	MAPE: 50.61	MAPE: 196.56	MAPE: 19.06	MAPE: 25.33

Table 5.5: TM performance with 3 types of normalisation.

5.2.2 With Volume

Volume integration into TM is a complex task that in some way goes against the original idea of transformers. Transformers were created to work with text data and sequentially predict the next character or word. If the Volume is added to the input stream, it is mixed with the closing price, making TM try to predict both. More complex data with many unique data points leads to a vast "alphabet" combined with the prediction of a not consistent data output stream will make TM highly inaccurate. TM is trained to predict the following number in sequence, and the appearance of 2 types of numbers in input data will be confusing for the model. In order to prove the theory explained above, we have implemented a one-week input and one-week output transformer, where the input is a mixture of Closing price and Volume. Volume and Closing price are normalised using Log scaling the same way it was performed for HDL models. The result of the experiment showed the following: MSE = 134.58, MAE = 8.89, RMSE = 11.60 and MAPE 122.66. As was expected, the model's accuracy significantly dropped compared to the one without Volume (see Table 4.5 Log scaling).

The overall results of all the models will be discussed in more detail in the next section of the report.

6. Results analysis

In this section of the report, the final result analysis will appear. It will discuss the main findings from experiments conducted and the models created.

Regarding HDL models, the input window experiment clearly demonstrated that out of all cases, the one week input window is the most beneficial to accuracy across the number of metrics when paired with one week or seven days output for both GRU-LSTM and LSTM-GRU.

Another important finding is that not all normalisation techniques work well with proposed models. Tanh estimator and Decimal Scaling were promising but failed to be helpful in our case. At the same time, Z-score, Min-Max and Log scaling were among the best tested formulas and satisfied all requests. They were chosen to be the best, and all further models used them and achieved good performance. However, among the three normalisation leaders, there was Min-Max that was the best for both Transformers and HDL models.

HDL models demonstrated great cryptocurrency price prediction power. Especially when input data is normalised using Min-Max or Log scaling, and the input window is seven days. However, regarding the Volume as an additional parameter, the results show that it affects HDL models in two ways (see Table 6.1). From the table below, it is clear that Volume, when applied to HDL models with Log scaling, improves the performance of LSTM-GRU and GRU-LSTM. However, when Volume is applied with Min-Max normalisation, the original model still outperforms the model with an additional parameter.

	Normalisation Method			
	Min-Max	Volume Min-Max	Log scaling	Volume Log scaling
LSTM-GRU	MSE: **0.0018**	MSE: 0.0025	MSE: 0.0214	MSE: **0.0199**
	MAE: **0.0305**	MAE: 0.0376	MAE: 0.1058	MAE: **0.1055**
	RMSE: **0.0424**	RMSE: 0.0503	RMSE: 0.1463	RMSE: **0.1413**
	MAPE: **11.28**	MAPE: 14.19	MAPE: 1.47	MAPE: **1.46**
GRU-LSTM	MSE: 0.0019	MSE: **0.00129**	MSE: **0.0253**	MSE: 0.0228
	MAE: **0.0314**	MAE: 0.0400	MAE: 0.1175	MAE: **0.1070**
	RMSE: **0.0438**	RMSE: 0.0542	RMSE: 0.1589	RMSE: **0.1513**
	MAPE: **11.71**	MAPE: 15.34	MAPE: 1.63	MAPE: **1.48**

Table 6.1: HDL Best Results With and Without Volume. Best results in **bold**.

Results for TM suggest some similar to HDL models conclusions. The input window experiment clearly shows that one week input is the best for Transformer; for more information, see Table 5.4. One week's input outperformed competitors by all metrics: MSE, MAE, RMSE and MAPE.

The normalisation experiment was conducted for TM in the same way as for all other models, and it did show that Min-Max normalisation and Log scaling both did great in increasing the performance results. However, there is no clear winner because many best metrics values are split between all three normalisations; for example, MSE is best in the case of Min-Max, but Log scaling outperformed every other one by MAPE. For the whole picture of the normalisation experiment, see Table 5.5, but generally speaking, Log scaling can be considered one of the best.

For the TM with Volume, there is a deviation from previously conducted data. The overall performance of TM suffered a significant decline across all evaluation metrics when the Volume parameters were added to the input. Even with one of the best normalisations, it still did perform extremely poorly (see Section 5.2.2 of the report).

TM demonstrated prediction power comparable to some of the implementations of HDL. In the table below, the best result before Volume is presented with the Volume result to clearly show the drop in prediction power (see Table 6.2). It is clear that the addition of Volume only harms the model's performance, and it is not to be considered a good additional parameter.

	Log scaling	Volume Log scaling
TM	MSE: **2.6672**	MSE: 134.58
	MAE: **1.3921**	MAE: 8.89
	RMSE: **1.6331**	RMSE: 11.60
	MAPE: **19.06**	MAPE: 122.66

Table 6.2: TM Best Results With and Without Volume. Best results in **bold**.

7. Conclusion

This last and final chapter of the report presents the overall conclusion for the HDL and TM models, a review of the initial aims set out at the beginning of this report, parts of the project that have been changed in comparison to the proposal, possible future work in the field of study and personal reflection on the project as a whole.

The overall conclusion is that Hybrid Deep learning and Transformer models were created and properly trained, and all valuable parameters were derived from the experiments or literature review. Volume as an additional parameter was concluded to be harmful for TM performance and positively affected HDL models. However, some interesting information was gathered in the progress of the study and will be further discussed in the future work part of this chapter.

7.1 Review of Aims

7.1.1 Hybrid Models

Aims: "*Create LSTM-GRU and GRU-LSTM hybrid models, train using ETH as the coin of choice and measure performance – to get the baseline data for the study.*" *& "Discover the optimal input window size and normalisation method for the Hybrid models."*

These first project aims have been successfully met. The hybrid models were created and trained in many formats: for input window experiment, for normalisation experiment. It allowed for determining the best input window, normalisation method and other additional parameters of HDL models. The best input window for this type of models was concluded to be a one-week input window, and the best normalisation to be Min-Max or Log scaling. The performance of the models at all stages of implementation is presented in the report in the implementation and results analysis sections.

7.1.2 Transformer Model

Aims: "Create a Transformer model suitable for time series prediction, train using ETH as the coin of choice, and measure the performance – to get the baseline data for the study." & "Discover the optimal input window size and normalisation method for the Transformer model."

For the Transformer model, all the implementation steps are discussed in the implementation chapter. All the initial goals regarding the transformer were met within the project. The designed model is suitable for ETH closing price forecasting. The performance was measured and compared to determine the best input window and best normalisation. The best input window was concluded to be one week input, and Log scaling or Min-Max to be the best normalisations for the model. However, the accuracy of the model was concluded to be lower than in case of Hybrid models.

7.1.3 Volume Addition

Aim: "Apply Volume as an additional parameter for presented models, experiment with different normalisation formulas, and measure models' performance after the input parameter changes."

Both models were changed according to additional input parameters in the form of the Volume of day trading for ETH. Different normalisations were applied to Volume in order to see if there was any significant performance difference. Log scaling normalisation appeared to be the best for HDL and the same is true for TM. Overall the conclusion was made that Volume as an additional parameter decreases the prediction power of TM and does affect HDL models slightly in a positive way. Overall, Volume is not recommended to be included in model training as an additional parameter.

7.1.4 Conclusion

Aim: "Make a conclusion based on collected data and present a verdict on the effect of volume as an additional parameter on hybrid deep learning models and Transformer."

At the end of the project, after all the experiments and comparisons, as was planned, the conclusion about the proposed idea was made. Hybrid models demonstrated great prediction power and accuracy. TM showed lower results but a clear potential for future improvement. Regarding the addition of the Volume of day trading of ETH, the conclusion was made that it is harmful to TM; however, it can be considered to improve the performance of HDL models slightly.

7.2 Changes in Comparison to Proposed Project

One of the main differences between this paper and the originally proposed project is that initially, it was expected to find three deep learning models previously implemented in studies for cryptocurrency price prediction and, later on in this study, to change these models to accept additional parameters of volume. However, in the progress of work, it appeared that no studies in the field shared the source code for the papers, or at least we were not able to obtain any models suitable for the study. Later the decision was made to implement the models within our study and make the changes to our models and compare the results before changes and after.

7.3 Future Work

In the progress of the research and experiment for this study, to the best of our knowledge, we utilised the Transformer model for time-series forecasting of cryptocurrency price for the first time. TM model demonstrated prediction power comparable to HDL, currently state-of-the-art models in time-series forecasting. It is the indicator that TMs have the potential to compete with HDL because in our study TM was smaller in size (345k parameters for TM to 405k for HDL), and still, it performed well for the relatively small size. Usually, TMs are much more significant; the number of parameters in GPT-3 was 175B [11]. It shows potential for improvement of TM by making it bigger.

Additionally, as suggested in [11], TM performs great when trained on a big data set and later fine-tuned to a specific task. Hence, one of the future research opportunities is to train a bigger Transformer on a data set consisting of, for example, the 100 most valuable cryptocurrencies and then fine-tune it for ETH closing price prediction. Such an approach will result in dramatic performance improvement.

7.4 Personal Reflections

As a whole, the project resulted in excellent skills developed in the fields of AI, Data Mining, Data Science and many more. Even before, the scope of my interests focused on cryptocurrencies, NFTs, technologies surrounding these areas and communities of people that develop new and innovative businesses. The developed project could especially help investors and crypto-communities if applied in a proper way and wrapped into easy to use service. In the progress of the project development, my personal skill set gained familiarity with the Google Colab platform, Python, TensorFlow, PyTorch and web scraping techniques. All these components could be beneficial in a number of fields of computer science, such as AI studies or Data Science.

Previously undertaken degree modules on Artificial Intelligence, Machine Learning, and Databases provided a solid basis for the work undertaken within this project. It presented an opportunity to improve my skills and apply the concepts taught and solidify my understanding. I now feel more confident in the areas of AI, ML and Data Science. Also, the skills gained will be helpful in future studies and career opportunities.

References

[1]
“Cryptocurrency: What is it and how does it work? - CBBC Newsround,” www.bbc.co.uk, Nov. 23, 2021. [Online]. Available: https://www.bbc.co.uk/newsround/57115261
Accessed on 20th October 2022.

[2]
D. Xi, T. I. O’Brien, and E. Irannezhad, “Investigating the Investment Behaviors in Cryptocurrency,” The Journal of Alternative Investments, p. jai.2020.1.105, Aug. 2020, doi: 10.3905/jai.2020.1.108.
Accessed on 16th October 2022.

[3]
CoinMarketCap, “Cryptocurrency Market Capitalizations | CoinMarketCap,” CoinMarketCap, 2019. https://coinmarketcap.com
Accessed on 24th October 2022.

[4]
E. Pintelas, I. E. Livieris, S. Stavroyiannis, T. Kotsilieris, and P. Pintelas, “Investigating the Problem of Cryptocurrency Price Prediction: A Deep Learning Approach,” Artificial Intelligence Applications and Innovations, vol. 584, pp. 99–110, May 2020, doi: 10.1007/978-3-030-49186-4_9.
Accessed on 24th October 2022.

[5]
A. M. Khedr, I. Arif, P. R. P V, M. El‐Bannany, S. M. Alhashmi, and M. Sreedharan, “Cryptocurrency price prediction using traditional statistical and machine‐learning techniques: A survey,” Intelligent Systems in Accounting, Finance and Management, vol. 28, no. 1, pp. 3–34, Jan. 2021, doi: 10.1002/isaf.1488.
Accessed on 23th October 2022.

[6]
S. Tanwar, N. P. Patel, S. N. Patel, J. R. Patel, G. Sharma, and I. E. Davidson, “Deep Learning-Based Cryptocurrency Price Prediction Scheme With Inter-Dependent Relations,” IEEE Access, vol. 9, pp. 138633–138646, 2021, doi: 10.1109/access.2021.3117848.
Accessed on 22th October 2022.

[7]
“Investing.com - Stock Market Quotes & Financial News,” Investing.com, 2019. http://investing.com

Accessed on 26th October 2022.

[8]
A. Politis, K. Doka, and N. Koziris, “Ether Price Prediction Using Advanced Deep Learning Models,” IEEE Xplore, May 01, 2021.
https://ieeexplore.ieee.org/document/9461061/metrics#metrics
Accessed on 22th October 2022.

[9]
T. Awoke, M. Rout, L. Mohanty, and S. C. Satapathy, “Bitcoin Price Prediction and Analysis Using Deep Learning Models,” Communication Software and Networks, pp. 631–640, Oct. 2020, doi: 10.1007/978-981-15-5397-4_63.
Accessed on 27th October 2022.

[10]
A. Vaswani et al., “Attention Is All You Need,” arXiv preprint, doi: 10.48550/arXiv.1706.03762, Dec. 2017.
Accessed on 10th February 2023.

[11]
T. Brown et al., “Language Models are Few-Shot Learners,” arXiv preprint, doi: 10.48550/arXiv.2005.14165, Jul. 2020. Accessed: Feb. 12, 2023. [Online]. Available: https://arxiv.org/pdf/2005.14165.pdf

[12]
Y. Son *et al.*, “Using Transformers and Deep Learning with Stance Detection to Forecast Cryptocurrency Price Movement,” *IEEE Xplore*, Oct. 01, 2022.
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9953018&casa_token=kIUD3_M0IHQAAAAA:NkOEJrjsYB1OXcxiFeWueEwdNw6LTPveuhl8MsbQNJrHBy-1m3f_Rmlef_mdmgHjALYDGhke7A&tag=1 (accessed Mar. 21, 2023).

[13]
A. Tanwar and V. Kumar, “Prediction of Cryptocurrency prices using Transformers and Long Short term Neural Networks,” *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*, Jul. 2022, doi: https://doi.org/10.1109/iciccsp53532.2022.9862436.

[14]
Yahoo Finance, “Yahoo Finance - Business Finance, Stock Market, Quotes, News,” @YahooFinance, 2009. https://finance.yahoo.com (accessed Feb. 13, 2023).

[15]
J. Sen and S. Mehtab, “A Time Series Analysis-Based Stock Price Prediction Using Machine Learning and Deep Learning Models,” International Journal of Business Forecasting and

Marketing Intelligence, vol. 6, no. 1, p. 272, 2020, doi: https://doi.org/10.1504/ijbfmi.2020.10038524.

[16]
I. Bordino, N. Kourtellis, N. Laptev, and Y. Billawala, “Stock trade volume prediction with Yahoo Finance user browsing behavior,” *IEEE Xplore*, Mar. 01, 2014. https://ieeexplore.ieee.org/abstract/document/6816733 (accessed Feb. 13, 2023).

[17]
S. Nayak, B. Misra, and H. Behera, “Impact of Data Normalization on Stock Index Forecasting,” *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 6, pp. 257–269, 2014, Accessed: Feb. 19, 2023. [Online]. Available: http://www.mirlabs.org/ijcisim/regular_papers_2014/IJCISIM_24.pdf

[18]
S. Bhanja and A. Das, “Impact of Data Normalization on Deep Neural Network for Time Series Forecasting,” *arXiv:1812.05519 [cs, stat]*, Jan. 2019, Accessed: Feb. 19, 2023. [Online]. Available: https://arxiv.org/abs/1812.05519

[19]
K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *arXiv.org*, Dec. 10, 2015. https://arxiv.org/abs/1512.03385 (accessed Feb. 22, 2023).

[20]
J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer Normalization,” *arXiv:1607.06450 [cs, stat]*, Jul. 2016, Accessed: Feb. 25, 2023. [Online]. Available: https://arxiv.org/abs/1607.06450

[21]
“LayerNorm — PyTorch 1.13 documentation,” *pytorch.org*. https://pytorch.org/docs/stable/generated/torch.nn.LayerNorm.html (accessed Feb. 25, 2023).

[22]
N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 2014, Accessed: Feb. 26, 2023. [Online]. Available: https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer

Appendices

Appendix 1

Deep Learning-Based Cryptocurrency Price Prediction in Relation to Trading Volume

Abstract

The proposed project reported in this paper will aim to investigate the effect of trading volume parameter on the performance of existing deep learning models for cryptocurrency market price prediction.

The project's first stage will involve finding deep learning models for price prediction of cryptocurrencies, as well as popular measures of prediction accuracy of such models. These models will be installed and run to collect baseline data that will later be evaluated using the mentioned earlier measures. Results will show the original predictive power of models. Later in the project, the models will be changed to take a volume of trading as one of the input parameters. Results from the project are expected to show that trading volume as a parameter is increasing the predictive power of deep learning models for cryptocurrency price forecasting.

1. Introduction

Cryptocurrency is a relatively new digital currency based on blockchain technology and cryptography. It has a decentralized and secure structure. Transactions made in cryptocurrency are nearly anonymous and provide freedom, as it allows users to make transactions without the need for a middleman, similar to banks for fiat currencies. All the transactions are stored in a publicly available blockchain database, which combines transparency and security [1].

The cryptocurrency market as a whole accomplished significant growth in the past decade. The market went from a single Bitcoin to thousands of currencies and tokens [2]. According to CoinMarketCap [3], there are over 21,000 cryptocurrencies, with a total market value of over $942 billion as of October 2022. The rapid growth made it very attractive for investors from all over the world.

Cryptocurrency popularity and volatility show the importance of prediction of the price. It is beneficial not only for individual investors as guidance for decision making, but also for financial researchers and studies that will be conducted in the future in the field of market behaviour. Price prediction for cryptocurrencies can be broken down as a popular time-series problem, similar to price prediction for stocks and anything with historical data available [4].

Cryptocurrencies do demonstrate non-linear patterns in price behaviour. When traditional machine learning tools are applied to cryptocurrencies, they perform imperfectly. Hence, there is a need for a more capable prediction tool, such as Deep Learning algorithms. It is a well-known solution for problems that involve forecasting complicated time-series problems. [5]

The proposed project will investigate the impact of the volume of trading on the predictive models, as only a few academic papers have considered this parameter. Furthermore, to our best knowledge, nobody measured the effect of trading volume on the hybrid deep learning predictive models. This will be carried out by testing the existing methods and applying a new parameter to previously tested models. The evaluation technique will be created to compare the predictive power of the models before and after the changes, and it will show the efficiency of the proposed idea. The information gained from the evaluation will allow for comparisons with previous studies of the cryptocurrency market price prediction.

The following part of the project proposal will be split according to the structure: background, the proposed project, programme of work, the resources required, and references. The background will contain a description of research projects previously conducted in the field of cryptocurrency price prediction, as well as methods and principles that were used before. Especially hybrid deep learning models, as it is the focus of the proposed project. The proposed project section will contain the main aims and objectives of this work. The programme of work will be a description of the plans for the project, including the breakdown of the scheduled tasks for the academic year in the form of a Gantt chart. The resources required section will focus on resources needed to complete the research successfully, and the references section will contain links and references to all the resources and academic papers used in this proposal.

2. Background

Previous projects relating to the proposed idea include deep learning-based cryptocurrency price prediction schemes with inter dependent relations [6]. The paper reports on how the use of parent coin data affects the hybrid predictive models. As the parent coin, Bitcoin (BTC) is used, because it is the oldest and most valuable coin on the market. Coins for prediction were Litecoin and Zcash, both are relatively old and historically replicate the movement of BTC market price. Public data for training and testing the models was taken from investing.com [7], which is a trusted source of information in the crypto community. For the prediction model two types of neural networks were used: Long short-term memory (LSTM) and Gated recurrent units (GRU), also flattening was used on the data of the parent coin. As mentioned previously models were combined into a hybrid model the output of which was combined with a flattening result overall producing a realistic and accurate prediction for windows of one, three, seven and thirty days.

However, this paper focused on the average price parameter and used it as the main information for hybrid deep learning neural networks, and so conclusively it is only showing the predictive ability with this limitation in place. Therefore, applying more types of publicly available data: open price, high of the day, low of the day, and the daily volume of trading; could show how the predictive ability of the model is changing depending on the input data. Such changes in predictive abilities in deep neural networks will be studied in the proposed project. Moreover, mean squared error (MSE) was used for the evaluation of the proposed model, and it does make it more complicated to compare to other papers in the field. Therefore, applying more popular evaluation technics could lead to more comparable results.

Furthermore, the Ether price prediction technics used by Politis et al. [8] also relate to the proposed project. The project involved price prediction for Ethereum (ETH), or Ether as it is referred to in the study. The idea of a parent coin was reflected in this study also, and the price of Bitcoin was used as the reflection of market dynamics. Also, the price of ETH and volume in USD were used, as well as some technical indicators: simple moving average (SMA) of two weeks, exponential moving average (EMA) for the same two weeks and Moving Average Convergence Divergence (MACD). Additionally, popularity indexes from Google trends for terms (ETH, Coinbase and Exodus) were used. However, as main characteristics were used two attributes of the ETH blockchain: daily block size and mining difficulty. For prediction in this study were used LSTM, GRU and Temporal Convolutional Network (TCN) models, also the different combinations of them in form of hybrid networks were tested: LSTM-GRU, LSTM-TCN, GRU-TCN. A comparison of the results for all six models made this study comprehensive. Furthermore, root-mean-square error (RMSE) and mean absolute percentage error (MAPE) were used to evaluate the results, which made the work easily comparable with other works in the field. Generally, hybrid models outperformed the individual ones, was concluded by the study.

This project showed how accurate the price of ETH can be predicted using different deep-learning neural networks and which hybrid models are more precise. However, data that was used as main characteristics are not relevant anymore to the Ethereum blockchain. A few months ago, ETH switched from proof of work (POW) protocol to proof of stake (POS). In the case of POS, the block in the blockchain does not have difficulty anymore, the number of blocks is fixed per unit of time. Hence, one of the main parameters is not usable anymore, it could be interesting to test the same network structures with a different and more relevant set of main characteristics. The proposed project will also involve the usage of hybrid neural networks but comparing it with other hybrids, not with individual models. And will try to focus on not out of date characteristics.

Another paper involved investigating the price prediction capabilities of deep neural networks in the case of cryptocurrencies [9]. As an object for the experiment, Bitcoin was chosen, as a market maker in the world of cryptocurrencies, prediction of the BTC price will give the overall view of the cryptocurrency market as all. LSTM and GRU models were chosen and tested, the goal was to determine the best one in terms of accuracy and compilation time. GRU outperformed LSTM in compilation time and overall demonstrated better ability in prediction for time series-related problems. Daily price values were used as a base of the dataset, including opening price, high price, low price, closing prices, and market capitalization for the day. The data was collected in the period from 2014 to 2018.

However, many daily price values were used in a data set, it was still not including the volume of trading. Also, data from only four years was used, that is not including the last four years of significant changes in the market. Therefore, it would be effective to repeat the project with an updated data set and more parameters in consideration.

As a result, this proposed project will specifically investigate hybrid neural networks for cryptocurrency price prediction. Hence, they are the most efficient ones, and not many experimentations with input parameters had been made, which makes it a challenging but interesting task to reinforce existing network models with volume data and discuss the results.

3. The Proposed Project

3.1 Aims and Objectives

The aim of the project is to investigate the impact of the daily volume of trading as an additional input parameter on hybrid deep neural networks used for cryptocurrency price prediction. Discuss the change in the predictive power of the models and make a conclusion. In order to collect the data suitable for the investigation, the next objectives will be utilized:

- Train existing hybrid models on the same data set – to get the baseline data for the study. It is essential to train different models on the same data set in order to get a comparable result. Later, the same testing data set will be used to evaluate the output of models. The specific coin and data to be used are specified in section 3.2.
- Modification of existing hybrid models concerning additional parameters of daily volume trading of a coin. Application of previously used data set to train new models. Once the models are changed and trained, algorithms will be tested on the same prepared data set as initial models from the first bullet point.
- Analysis of the data from existing and changed models – When both original and modified models are trained and tested, the final prediction output will be evaluated using the same parameters. Moreover, the data will be analyzed to determine whether the trading volume is valuable for cryptocurrency price prediction. This will be discussed in the conclusion of the paper.

3.2 Methodology

For the research in the proposed project, a sample of 3 existing hybrid neural networks will be taken. The models should originate from the existing studies in the field to be as relevant as possible. Moreover, should be re-trained on the same data set. The usage of existing algorithms that are trained on a similar data set will allow for the confident and reliable result of the evaluation of the predictive power of the models. Chosen networks must be trained and tested on the same data set from one cryptocurrency. Taking into account the fact that many possible models use parent coin data of BTC cannot be used for training, another currency needs to be chosen. Most probably it is going to be ETH, as it is an old and tested blockchain with great potential, historically it is following the trend of BTC price, and it is needed for many models with a parent coin concept in place.

After training and testing the existing models, all three models will be changed according to the need of the project. The additional layers of neurons will be added to include the effect of day trading volume on the forecasting of the future price. Modified hybrid models will be trained on the same data set as the old ones and adequately tested on the same test set to return a reliable result for evaluation.

For evaluation purposes, the most popular technics will be used. It does include the RMSE to measure the difference between observed values predicted by the model and MAPE as an additional measure for forecasting methods. Many studies in the field use these methods, so the proposed project will be easily comparable to other papers.

Due to the aims and objectives of the project, there will be two-phased evaluations; the first phase involves training, testing and evaluating the original versions of deep neural networks, and all the same is done for new networks. The second phase will consist of the evaluation and comparison of the results of the first two evaluations; it will allow to compare the study results properly.

How the project will be performed is described in the next section Programme of Work.

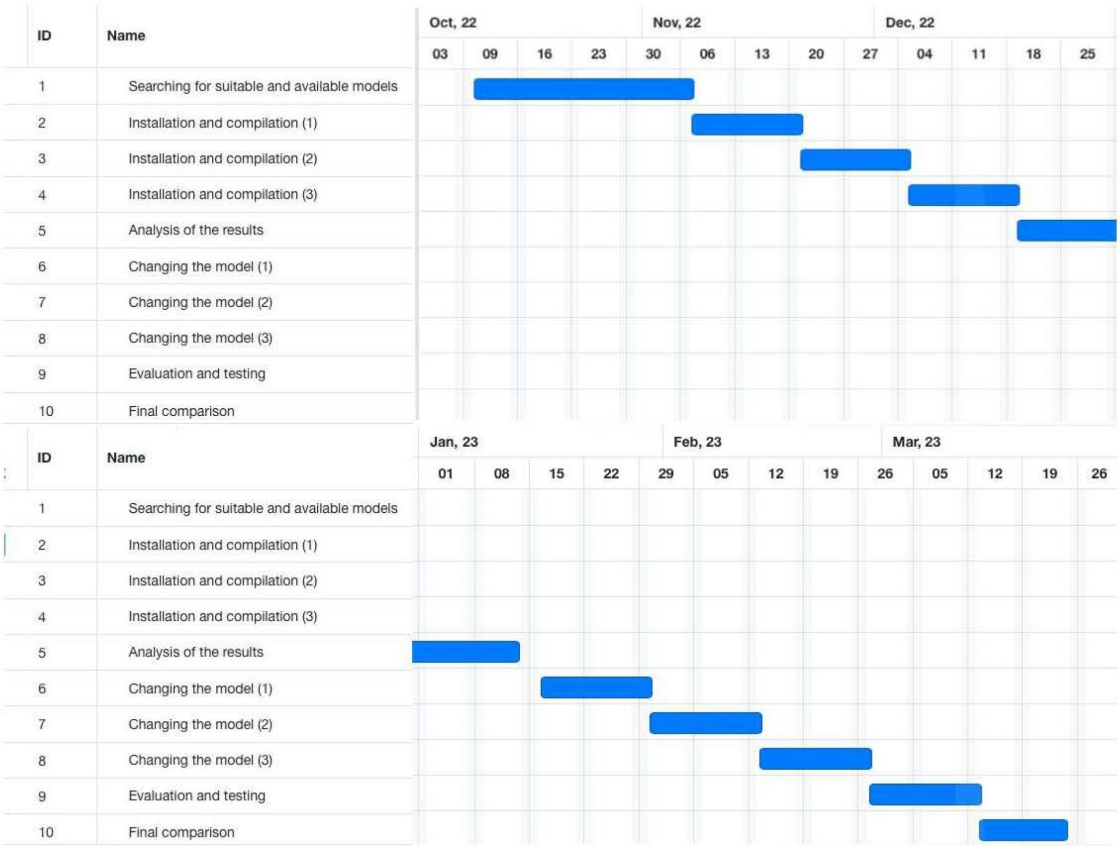
4. Programme of Work

The project will begin in early October 2022 and continue until March 2023, and it will be split into the following stages:

- Searching for suitable and available models – This will involve reaching out to researchers in the field and asking them for code related to the published papers, searching the web and GitHub repositories for the relevant original code. The goal is to find the code for hybrid predictive models based on deep neural networks used for cryptocurrency price prediction. The aim is to find three such models with available source codes. It is an important step, and it will take four first weeks of the project time.
- Installation and compilation – This step will contribute to each of the found algorithms. Also, the evaluation metrics will be applied to the output of each model. It is essential to install and run all models on the same machine to get reliable data for evaluation. Approximately it will take two weeks for one algorithm to go through installation and compilation.
- Analysis of the results – Analysis of the evaluation results from the previous step will result in the baseline performance of the algorithms, trained and tested using a new data set. The performance of all three models will be summarized and compared. This stage will take 3-4 weeks during the Christmas break.
- Changing the model – This stage will be applied to all three used algorithms previously. It will include changing the model, so it relies more on ETH's volume of day trading. New networks will be trained and tested using the same data test as in the previous steps, ensuring a truthful comparison. This will take around one to two weeks per model.

- Evaluation and testing – During this stage, new models will be tested using the same evaluation techniques applied in the project's second stage. It will show a real improvement in the performance of the models. This will be carried out in the following two weeks after the new models' creation.
- Final comparison – Once all the data is collected. Old and new models are trained and tested, and all the data is evaluated using the same metrics; it is time to make the final comparison. In this stage, data from all previous steps will be evaluated to decide whether the models' predictive abilities are improved within new data and settings. This part of the project will take around two weeks.

The overall plan for the proposed project for 2022 – 2023 is displayed by a Gantt chart in Figure 1 below.



5. Resources Required

Access to Google Colab, as many projects in machine learning developed in this collaborator from google for python code which is especially suitable for machine learning. This platform will make working on code easy, mainly if the initial code was written using it, and it will make execution or edition of code possible on any machine. Also, HEC graphics processing unit (GPU) might be needed to compile and run the prediction models in a reasonable time frame. Lancaster University will provide the services mentioned above.

6. References

[1]
"Cryptocurrency: What is it and how does it work? - CBBC Newsround," *www.bbc.co.uk*, Nov. 23, 2021. [Online]. Available: https://www.bbc.co.uk/newsround/57115261
Accessed 20th October 2022.

[2]
D. Xi, T. I. O'Brien, and E. Irannezhad, "Investigating the Investment Behaviors in Cryptocurrency," *The Journal of Alternative Investments*, p. jai.2020.1.105, Aug. 2020, doi: 10.3905/jai.2020.1.108.
Accessed 16th October 2022.

[3]
CoinMarketCap, "Cryptocurrency Market Capitalizations | CoinMarketCap," *CoinMarketCap*, 2019. https://coinmarketcap.com
Accessed 24th October 2022.

[4]
E. Pintelas, I. E. Livieris, S. Stavroyiannis, T. Kotsilieris, and P. Pintelas, "Investigating the Problem of Cryptocurrency Price Prediction: A Deep Learning Approach," *Artificial Intelligence Applications and Innovations*, vol. 584, pp. 99–110, May 2020, doi: 10.1007/978-3-030-49186-4_9.
Accessed 24th October 2022.

[5]
A. M. Khedr, I. Arif, P. R. P V, M. El-Bannany, S. M. Alhashmi, and M. Sreedharan, "Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey," *Intelligent Systems in Accounting, Finance and Management*, vol. 28, no. 1, pp. 3–34, Jan. 2021, doi: 10.1002/isaf.1488.
Accessed on 23th October 2022.

[6]
S. Tanwar, N. P. Patel, S. N. Patel, J. R. Patel, G. Sharma, and I. E. Davidson, "Deep Learning-Based Cryptocurrency Price Prediction Scheme With Inter-Dependent Relations," *IEEE Access*, vol. 9, pp. 138633–138646, 2021, doi: 10.1109/access.2021.3117848.
Accessed on 22th October 2022.

[7]
"Investing.com - Stock Market Quotes & Financial News," *Investing.com*, 2019. http://investing.com
Accessed on 26th October 2022.

[8]
A. Politis, K. Doka, and N. Koziris, "Ether Price Prediction Using Advanced Deep Learning Models," *IEEE Xplore*, May 01, 2021. https://ieeexplore.ieee.org/document/9461061/metrics#metrics
Accessed on 22th October 2022.

[9]
T. Awoke, M. Rout, L. Mohanty, and S. C. Satapathy, "Bitcoin Price Prediction and Analysis Using Deep Learning Models," *Communication Software and Networks*, pp. 631–640, Oct. 2020, doi: 10.1007/978-981-15-5397-4_63.
Accessed on 27th October 2022.

Appendix 2

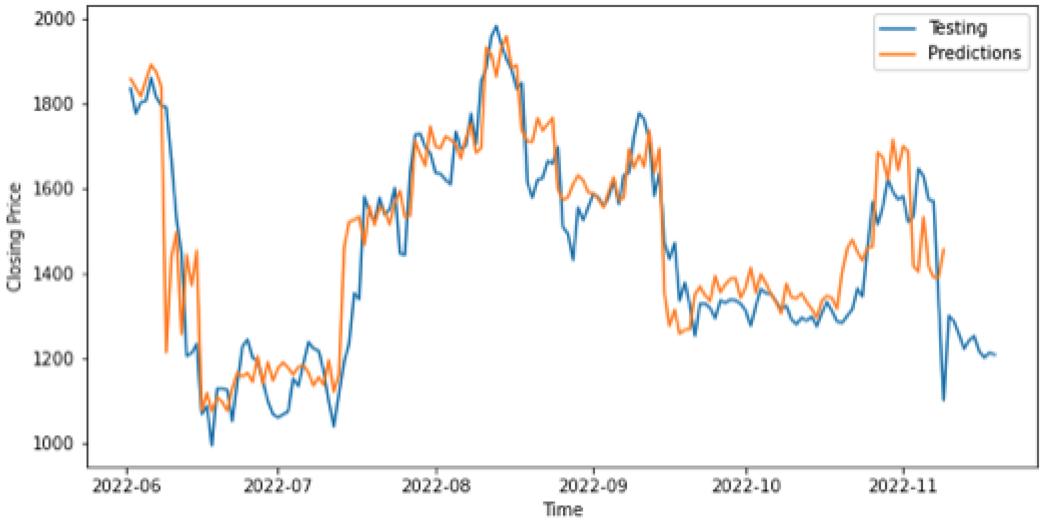


Figure A2.1: Example of prediction graph from the study (GRU-LSTM)

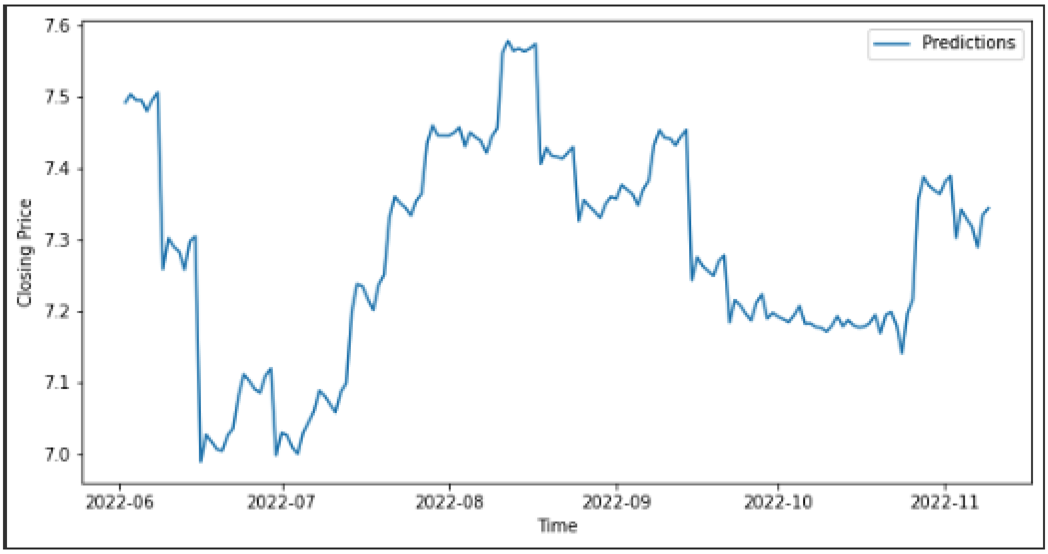


Figure A2.2: Example of prediction with Log scaling normalisation (LSTM-GRU)