Univariate and Multivariate Time Series Forecasting in Ethereum

Ramakrishna Puttala Karun Kumar
MSc Big Data Science
Emmanouil Benetos
210542309

***Abstract*—The Cryptocurrencies are one of the most dynamic markets introduced recently to the financial ecosystem and available for traders. Over 40-billion-dollar worth of Cryptocurrencies are traded every day. In a recent survey, it has been observed that these are the second largest market in terms of Average Daily Trading Volume. It is notoriously known for its volatility, accessible to few lucky well timed opportunists, while others having faced suffered devastating losses due to poor judgements. If these price movements can be predicted in advance to obtain rewarding result with proper risk management by the aid of the known statistical models like ARIMA, Machine Learning models (Linear Regression, Decision trees, SVR, etc.) and state-of-the-art Deep Learning models (LSTMs, GRUs etc.) which is the main objective of this paper. This paper also aims to assess the Deep Learning models performance to other ML models in domain of forecasting by analyzing the various key performance metrics of these models. We are specifically interested in the actual target prediction with these models. In this paper we also explore the affect of resampling in reducing the error towards the actual target prediction. There is a raw (OHLCV) data from which we need to extract basic and advanced features. The paper elucidates the ease of models while prediction of linear trends. However, the deep learning models were able to deal better with complex predictions than others.**

***Index Terms*—Machine learning, Deep learning, Finance, Ethereum target prediction, time-series forecasting.**

I. Introduction

Cryptocurrencies are the millennial gold. Machine learning (ML) and Deep Learning (DL) has now become a sophisticated analytical tool for managing investments effectively in the financial markets. Unlike equities, Cryptocurrencies rise in the price can go over and beyond 100 % in both directions just in a day and that's one of the reasons for popularity that everybody eagerly involved in Crypto market. Cryptocurrencies can give investors handsome rewards when they are infallible, however on the downside, their unpredictive movement can wipe out entire capital of investors when they are holding a position on the wrong side [1]. In order to improve their chances of generating larger returns, investors must therefore grasp the characteristics, fundamentals of various crypto assets and the factors that affect their respective prices. However, despite of this investors still have the liability to do smart investments at the precise moment utilising reliable and pertinent information, such as market mood [2].

Are these prices predictable? There are multiple theories stating that the price movements follow a random walk process [3] [4], semi-random or predictable [5] [6]. In earlier sections of this paper goes on the variations from the mean on each day are random and unpredictably distributed. In later sections, the paper considers that markets are somewhat predictable and that the random walk theory of stock prices is incorrect.

Time series data are collections of continuous data collected over constant time intervals(T). Data is collected on a yearly, monthly, weekly, daily, or hourly,minutely, or secondly basis. Time-series data is comprised of mainly four components: level, trend, seasonality, and Irregularity. Level is the series' baseline value if it were a straight line. Trend is the series' optional and often linear increasing or decreasing behavior over time [7]. Seasonality is Repeating patterns or cycles of behavior that can occur over time. Irregularity is the optional variability in the observations that the model cannot explain. Every time series has a level, the majority have noise, and trend and seasonality are optional. Most Time-Series (TS) models are based on the assumption that the TS is stationary. By stationary we mean three things:

1) Mean constant over time.
2) Variance constant.
3) Co-variance is only a function of gap.

The reason why we need stationarity in the time series data is because we learn a bunch of statistical properties from the data at hand and it's helpful for forecasting. Just like if we know that the distribution is Gaussian we tell a bunch of properties about the data distribution. We can either use differencing or Log transform to make it the time series stationary.

Machine learning (ML) is acting as a major technique in a variety of critical applications. Among ML algorithms, Linear regression, Decision trees, Random Forest, Support Vector Regression etc.., are selected. The hyperparameters for these models were set by the grid search algorithm. The performance metrics used to assess these models were MSE [8]and MAPE [9].

Recurrent Neural Networks (RNNs) are kind of powerful Deep Learning (DL) model for processing sequential data such as sound, time series data, or written natural language [10]. Some RNN designs have been used to forecast the stock market [11]. One of the most successful RNN architectures is Long Short-Term Memory (LSTM) [12]. LSTM introduces cells, allowing them to grasp the structure of data dynamically over time with high prediction capacity. Unlike LSTM, Gated Recurrent Units (GRU) is a different architecture of RNNs which is faster to train than a LSTM for target prediction. Both LSTM and GRU the historical data of Ethereum were transformed into 15 min long sequences with 11 learning features and 16th min return is used as target. The model was fitted by training on 1369491sequences and tested using the other 586901sequences.

In this paper, we propose to investigate the possibility of tracking and predicting the returns of Ethereum. The purpose of this paper is to investigate the feasibility of using a machine learning and deep learning mechanism in the Crypto market especially Ethereum target prediction and to compare the results of actual versus predicted prices by resampling the data into different time intervals i.e. (1min,5min,15min and 1Hr) using a machine learning and deep learning algorithm.

The paper initially briefly discusses the previous work and literature survey performed on the field of problem statement delving on various models such LSTMs, SVR, ARIMA, Linear regression etc., Then the next section explains the problem statement and limitations, challenges that it presents. An approach to solve the problem is proposed in the later section. To solve the problem in the proposed approach, methodology of solution is presented in the section with steps such as Dataset, Data Pre-processing, feature extraction and so on till the model evaluation. The later section discusses on the results obtained from the empirical findings of each model following the methodology and their comparisons with performance metrics tabulated. Finally, providing the conclusion to the models solution and results with the insights from the previous works and future work that can enhance the solution further.

II. Related Work

There are lot of research papers that had focus mainly on the financial instruments (namely Stocks, Bonds, commodities, Derivatives and Forex) but a very few that focuses on the Cryptocurrencies. However, the research and analysis of these financial instruments can be extended to the highly volatile market Cryptocurrencies as well. There are various Statistical models, ML and DL algorithms and techniques being applied to these financial instruments. Now, we mainly focus on the actual target price prediction which is a regression task.

A. Autoregressive integrated moving average models(ARIMA):

ARIMA models which have an underlying assumption that the series are generated from a linear process and results might not be appropriate for the majority of non-linear real-world cases [13].

These papers applied the ARIMA on Dell stock using OHLCV data to predict the stock price and compared the results of the Artificial neural network(ANN). The empirical results for ARIMA infers that it is effective for the stock price prediction. Additionally, the ARIMA forecasting models have a directional pattern which was observed during their experiments. On comparing the results of both models ANN has given better predictions over ARIMA [14].

B. Linear regression (LR) [15]:

Linear regression is a technique to estimate the dependent variable by one or more independent variables.They used 30 financial indicators to predict the Stock price in Tehran stock exchange [15]. It compared the results between Linear regression and Artificial neural network (ANN) models. And their experimental results showed that the Neural network is good at estimating than the regression models. However, Linear regression models are affected by the outliers in the data.

C. Decision Tree (DT), Random Forest (RF) and Gradient Booted Trees (GBT):

Decision trees are extensively used in the field of finance because they can solve the problem of forecasting non-linearity and non-stationary time series dataset.

Random Forest (RF) is a machine learning algorithm used extensively for classification and forecasting tasks. It can handle time-series and non-linear data. Robust method for forecasting and classification tasks due to its design, which is filled with various decision trees and the feature space is randomly modelled.

RF requires more computational power and resources because it generates a large number of trees. Takes longer to train than decision trees [16]. Their research was to use the various tree-based methods and traditional statistical methods like ARIMA for forecasting monthly gold prices. Their experimental results showed that based on the RMSE (Root mean squared error) the Random Forest (RF) outperformed various tree-based algorithms (DT and GBT) and statistical models.

GBT is one of the most effective methods for creating predictive models for both classification and regression tasks is gradient boost. Gradient boost sequentially joins numerous weak learners to create a strong learner, property of boosting models. Decision trees are frequently employed by Gradient boost as poor learners. It creates a strong tree by sequentially joining weak trees to increase the model's accuracy, achieving low bias and variance. The distinction between GBT and random forest models is the order of the trees that are constructed. The fit is improved with each iteration by the subsequent tree fitting the residual errors from the preceding tree.

D. Support Vector Regression (SVR):

It is used for financial time-series Forecast modelling and known to be powerful for predicting financial time-series, handling the high-dimensional data and giving high prediction accuracy capable enough to deal with overfitting [17]. Sensitive to free parameters that are set by the user being adaptive. The paper proposed combination of linear regression and SVR for time series analysis. That is divided to linear and non-linear models respectively. However, the paper considered only first order linear model for stable part.

E. Neural Network

Neural Network (NN) It was found in 1989 that neural networks (NNs) are effective and perfect for forecasting and categorizing data of financial markets as they can spot patterns in huge chaotic data and can handle non-linearity [18]. There are various Neural Network architectures of which one is Multi-Layered Perceptron (MLP) is the simplest and widely used architecture used with back-propagation algorithms mostly used in the stock market analysis. Even with several drawbacks such as being too slow for the real-world applications. To overcome, more advanced Neural Network like RNNs variants like LSTMs and GRUs are introduced to analyse stock market [19].

F. LSTM Long Short Term Memory

The field of financial time series forecasting is dominated by LSTM and its variants, as well as a few hybrid models. Financial time series forecasting is a well-researched and successful because LSTM, by its very nature, makes use of the temporal properties of any time series signal. However, some researchers prefer to either extract the necessary features from the time series or transform the time series so that the resulting financial data become stationary in terms of time, meaning that even if the order of the data is changed, the model can still be successfully trained, and the out-of-sample test results will be accurate [20].

GRUs are Deep learning models unlike LSTMs they only have two gates and faster to train than LSTMs. In [21] the raw OHLCV data from Yahoo finance API and predicted the future price of the intel stock price by using GRUs with slight modification in the internal structure and they used RMSE as a metric to predict the future price of that stock better accuracies.

The literature review from Table. I shows some models perform better than other models, with different features for the financial instruments in the task of time series forecasting. But, to the best knowledge there are aren't any papers that focuses on resampling the data, Univariate or multivariate time series forecasting in case of cryptocurrencies that helps in better performance accuracies or price predictions using technical indicators. This paper therefore seeks to check these objectives with just the technical indicators discussed in the feature extraction part of this paper. Results obtained are based on the experiments performed on the Ethereum data taken from Kaggle.

Model / Algorithm	Analysis
ARIMA and Linear Regression	Based on empirical findings ARIMA and Linear regression tend to determine the linear relationship between the features and hence good at identifying the trend of the data which is the price variations. But often fail with non-linear data. So, for our analysis we build these models to keep them as baseline models and compare it with the other various model's performance [7].
Tree based methods (DT, RF and GBT)	Based on the research these models have the ability to work very well with non-linear time series problem. But tend to easily overfit in case of outliers. They are fast to train and can be used in the case where there is a latency issues, because we store a just the if and else conditions [22].
SVR	Coming up with the right kernel for the task is cumbersome as it requires lot of training and testing after finding the right kernel solving the problem is easy task. They remain one of the favourite models because of auto featurisation. They can tackle overfitting with ease. Sensitive to the outliers [17].
LSTMs and GRUs	LSTMs and GRUs are Deep learning models that are used for the financial time-series forecasting problem and are state-of-the-art techniques. They are capable of analysing and exploiting data interactions and patterns through a self-learning process. Makes accurate predictions because it analyses data interactions and hidden patterns. Good at remembering information for a long period of time. They work very well with non-linear time series data and are capable of handling outliers. But often can be overfitting to the data [23] [24].

TABLE I: *Comparison table between all models*

III. Problem Statement

Feature extraction and training the model with the right features and coming up with right size for the moving window are the challenges in establishing an accurate target prediction. The existing ML models and DL models are trained on the dataset. The goal of this project is to build a model which accurately predicts the target.

This project is based on four main objectives:

- Actual target prediction using the various ML and DL models.
- Resampling the data into different time frames [15min, and 1D] and explore to see if resampling the data into intervals helps in reducing the error in the target prediction using various ML models.
- Performing the Univariate time-series forecasting using a moving window and Multivariate forecasting in order to compare if Univariate is doing better than multivariate time series forecasting and vice-versa.
- Explore to see if the Deep Learning models outperform Machine learning models based on the performance metrics.

The outliers in the data can be very important due to volatility of cryptocurrency being high, and that's the reason for not discarding any outliers. The main argument senario is that if a machine learning and deep learning models doesn't predict a sudden rise or fall in the price of the cryptocurrencies which is meant to be the outliers in the data. We explore to see the behaviour of these models in the presence of the outliers. Various ML and DL models have been trained and tested on the dataset with changes made to those models for better performance. The implementation of Univariate and Multivariate time-series forecasting with their detailed analysis.

IV. METHODOLOGY

A. Dataset

The data set was taken from Kaggle [25] that was part of the competition for Time series forecasting hosted by G-Research and this dataset is publicly available on Kaggle. Historical timeseries data for any financial instrument is not confidential and can be acquired from many platforms using the APIs. This dataset contains historical trade data for several crypto assets, including Bitcoin and Ethereum. Of which we are only interested in Ethereum. Ethereum data goes from 2018-01-01 to 2021-09-21. The data set contains 1956200 data points. The dataset has following features:

- timestamp- A timestamp for the minute covered by the row.
- AssetID- An ID code for the crypto asset.
- Count- The number of trades that took place this minute.
- Open- The USD price at the beginning of the minute.
- High- The highest USD price during the minute.
- Low-The lowest USD price during the minute.
- Close- The USD price at the end of the minute.
- Volume- The number of crypto asset units traded during the minute.
- VWA - The volume weighted average price for the minute.
- Target- 15 min residualized returns.

Using these features, we need to predict the Target.

B. Pre-processing

The basic study and analysis of the Ethereum dataset showed there were missing values in the Target column and this dataset has many gaps between the successive rows in timestamp attribute. When performing time-series analysis there should not be any inconsistencies such as time-gaps and if present, then information is lost, and performance is limited. The workaround to mitigate this problem was to reindex so that the data has none of the time gaps, and each successive row is at a frequency of 60s. The missing data in the target feature has been fixed with the spline interpolation technique which calculates the values that produce a smooth surface going through the input points by minimising the surface's overall curvature. Min-max Feature scaling is applied on the entire dataset to scale the features between 0 and 1. And, for resampling the data into different time frames[15min,1D], the data aggregation has been implemented using pandas module.

C. Performance Metrics

The performance metrics for the evaluation of the statistic models, Machine Learning models, and Deep Learning models for our task are:

I. Mean Squared Error: This metric is not interpretable. And is extensively used as a metric to see how the regression models are performing [8].

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$\text{MSE} = mean\ squared\ error;\ n = number\ of\ data\ points$$

$$Y_i = observed\ values\ ; \hat{Y}_i = predicted\ value$$

II. Mean absolute Percentage Error (MAPE): This metric is interpretable. However, it tends to deviate towards indefinite. So, to overcome this limitation modified version of MAPE which is taking the mean of actual value instead of just taking the actual value [9].

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{Y}_i|}{Y_i} \times 100$$

$$Y_i = observed\ values\ ; \hat{Y}_i = predicted\ value$$

MAPE is good metric compared to MSE because it a relative error. In the case of MSE, we will only get the difference.

D. Feature extraction

The Feature extraction is the important step in the Machine learning and deep learning. The good the features the better the model prediction. For our problem we have taken the following features.

- Close Difference: It's the difference between the close price of Ethereum between the successive timestep.
- Moving Average: A moving average (MA) is a popular stock indicator in technical analysis. To smooth out price data over a specified period of time by creating a constantly updated average price. To analyse the trend of the financial securities. The ones that were used for the featurisation are MA200, MA100, MA50 which are used to see the long-term trend while MA26, MA20, and MA12 are used to see the short-term trend [26].
- Moving average difference: It's the difference between different moving average. The ones used for the featurisation are Difference between MA200 and MA50, Difference between MA200 and MA100, Difference between MA200 and Close price of Ethereum, Difference between MA100 and Close price of Ethereum, Difference between MA50 and Close price of Ethereum [26].
- Exponential Moving average: An exponential moving average (EMA) is a kind of moving average (MA) that gives the most recent data points more weight and significance. To analyse the trend whether a financial security is bullish or bearish. Several EMAs are used for example: EMA12, EMA20, EMA26, EMA100, and EMA200 [26].
- Shifts in the close price of Ethereum on previous timestep and 2 previous timesteps.
- Bollinger Bands: It's the technical indicator often used by traders. The upper and lower bands are typically two standard deviations plus or minus from a 20-day moving average, but they can be adjusted. They tell about the overbought and oversold regions [27].
- Oscillator-type indicators indicate the possibility of reversing a current trend. This category includes indicators such as moving average convergence/divergence (MACD) [28] and relative strength index (RSI) [29], among others.
- Momentum Type-These indicate the rate of change of a security's price. Momenta with various window sizes are included.
- Volatility Indicator- It's the difference between the High and Low price of Ethereum divided by the VWAP [30].
- Trend Indicator- It's the difference between Close and Open price of Ethereum divided by the VWAP.

- HL- It's the difference between the High and Low price of Ethereum. It captures the fluctuation in the price for a given day.
- K-ratio – It's the percentage of difference between the Close and MA14 on Low price of Ethereum divided by the difference between MA14 on High price of MA14 on Low price of Ethereum.

By performing feature extraction the dimensionality of the data increased to 46 features and the highly correlated features were discarded using pearson correlation [31] reducing the dimensionality to 9 features.

E. Training various models

1. **Statistical Models**

Auto ARIMA: The dataset was resampled into [15min,1D] and then performed a stationarity test using the augmented dickey fuller (ADF) test on the Target feature. The results showed that the Target feature is stationary. Later performed the Auto ARIMA on the Target feature to get the p, d, and q values for training an auto ARIMA model. For both 15min and 1D dataset, the values for p and q are set to $p = 10$and $q = 10$.

The 15 min and 1D dataset used to train the auto ARIMA model was split into 70% training and 30% test sets. The auto ARIMA model was fit on the training data and tested on the test data using the parameters resulted from auto ARIMA model. And the performance of this model is tabulated in the Table. II.

2. **Machine Learning Models**

I. *Univariate Time-series forecasting using a moving window:*

The dataset for both 15min and 1day was transformed into moving window of 5 i.e., taking the last 4 timestep $(t - 4)$ to prediction the next timestep $(t + 1)$. Refer Fig.1 for better understanding.

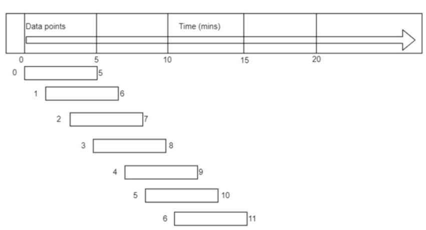

Fig. 1: *Moving Window Process*

The dataset was split into the 70% train and 30% test sets.

If the dimensionality of the data increases time complexity to train DT also increases. Logical feature interactions help in deciding the class labels and are in-built into DTs. As depth of the DT increases the impact of the outliers can be significant and tree becomes unstable. Decision trees are super interpretable when the depth is reasonable.

Various Machine Learning models namely Linear regression, Decision trees, Random Forest, Support vector machine regressor and XGBoost Regressor were trained and tested and hyperparameter tuning is performed using Grid search CV. And their performance is tabulated in the Table. III, IV.

II. *Multivariate Time-series forecasting:*

The dataset for both 15min and 1day with the features mentioned in the feature extraction part in the section D were used along with the raw dataset. We performed the pearson correlation [31]to check if there are any of the correlated features in the data set and then the removal of these correlated features were done. After removal of the correlated features, we trained various Machine Learning models namely Linear regression [15], Decision trees, Random Forest [16], Support vector machine regressor [17] and XGBoost Regressor were trained and tested and hyperparameter tuning is performed using Grid search CV to get the best parameters for training these models. And their performance is tabulated in the Table. III, IV.

V. Deep Learning Models

A. Data Preparation

Data Preparation is not trivial in the case of Deep Learning where obtaining the data in the right format is the most crucial task. It is fed to the LSTM cells as a tensor. So, for the univariate time series forecasting the data is split into 70% train and 30% test set. And then using the last 15 previous values from the target column we are trying to predict the 16th target value for the entire data frame. Train data is of the shape (1369491,15,1) and test data is of the shape (586916,15,1). Fig. 2 showcases the pipeline of building a LSTM model.

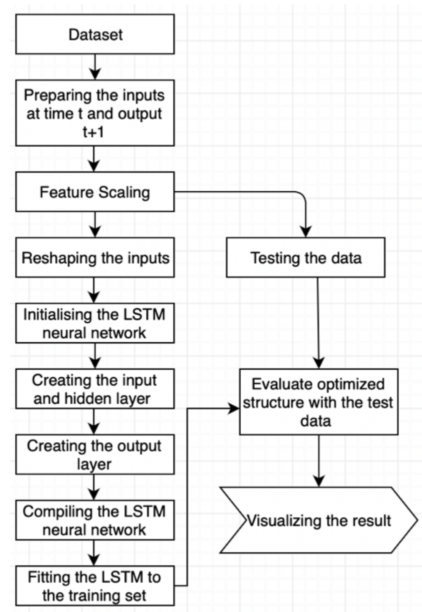

Fig. 2: *Pipeline for building a LSTM model*

For the Multivariate Forecasting single step and multistep forecasting, we have used the same features from that were discussed in the section D. After removing the correlated features, Multivariate Forecasting single step was implemented using the last 15 previous values from the target column we are trying to predict the 16th target value for the entire data frame using all the non-correlated features as well. So that resultant train data is of the shape (1369491,15,11) and test data (586916,15,11).

Multivariate multistep Forecasting was implemented using the last 15 previous values from the target column we are trying to predict the next 15 target value for the entire data frame using all the non-correlated features.

B. Model Architecture

For the forecasting problem, we used LSTMs and GRUs for this task.

LSTM: For univariate time series forecasting LSTM model is built using 2 layers of LSTM. The First LSTM layer consists of 64 cells and the second layer consists of 32 cells both using the relu activation function. I have set the dropout to be 0.2 and output layer is a dense layer consisting of 1 neuron. I used Adam optimiser and loss function was MSE.

For multivariate time series single step forecasting LSTM model is built using single layer of LSTM and it consists of 8 cells. I have set the dropout to be 0.2 and output layer is a dense layer consisting of 1 neuron. I used Adam optimiser and loss function was MAE.

For multivariate time series multistep forecasting LSTM model is built using 2 layers of LSTM. The First LSTM layer consists of 16 cells and the second layer consists of 8 cells both using the relu activation function. I have set the dropout to be 0.2 and output layer is a dense layer consisting of 15 neurons to give out the predictions for the next 15 min time frame. I used Adam optimiser and loss function was MAE.

The similar architectures are implemented for the GRUs as well.

C. Training the LSTMs and GRUs

Different epochs and batch sizes were considered during the training of these models and the best model which gave the best results i.e. the one with the lowest error was used to predict the test set and the results are tabulated in the Table. V.

D. Evaluation

After training the models, we test the model with the test dataset. We evaluate the models using the performance metrics discussed in part B.

VI. Results And Discussion

We used Jupyter notebook for our analysis, training, testing and evaluation of the models. And the results are tabulated in the following Table.II below.

Results for auto ARIMA:

The results after training univariate ARIMA model showed that the for 1 Day data the parameters for autoregressive(p) and moving average (q) are ARIMA (3,0,0). And for 15 min data the parameters for autoregressive(p) and moving average (q) are ARIMA (0,0,8). And the performance is tabulated in the Table. II and Fig. 3, 4.

Dataset	MSE		MAPE	
	Train set	Test set	Train set	Test set
15 MIN	0.000466	0.000678	3.000	3.702
1 Day	0.00264	0.00589	6.269	9.230

TABLE II: *The results for ARIMA time series forecasting*

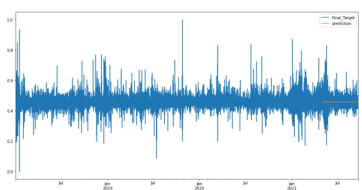

Fig. 3: *The predictions of ARIMA for 15 min data set*

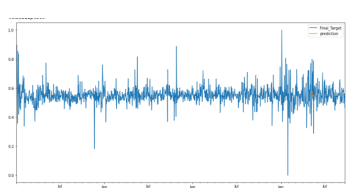

Fig. 4: *The predictions of ARIMA for 1 Day data set*

As can be seen from the Table. III, for 15 min univariate and multivariate time series forecasting, the Decision trees, Linear regression and SVR and XGBoost gave the best results for univariate forecasting and Decision Trees and Linear regression gave the best results for multivariate forecasting.

The MAPE error for Decision tree univariate forecasting is 2.997% in both positive and negative direction for the train set and 3.707% in both positive and negative for the test set. The MAPE error for Decision tree multivariate forecasting is 2.978% in both positive and negative direction for the train set and 3.703% in both positive and negative for the test set. Also, it's worth noting that for univariate forecasting the linear regression,SVR, and XGBoost regressor refer Fig.10 are performing almost the same in comparison with the Decision trees. But, for multivariate forecasting Linear regression is also doing a better job at forecasting. The XGBoost Regressor for the same univariate can be referred to Fig.6.

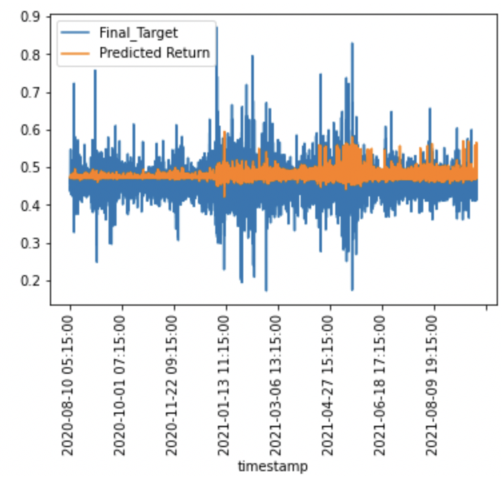

Fig. 5: *Multivariate Forecasting with XGBoost Regressor for 15min*

From Table. IV, It can be observed that by resampling the errors in the forecasting increased because of re-sampling data reduced. To choose a model based on the empirical results, the Decision trees and Linear regression be higher, with cost of overfitting in both the cases. Error is not so huge for train and test, seems acceptable for forecasting scenarios. The overfitting is maybe due to the train and the test data having different distribution with being resampled for 1D univariate and multivariate time series forecasting.

Fig. 6: *Univariate Forecasting with XGBoost Regressor for 15min*

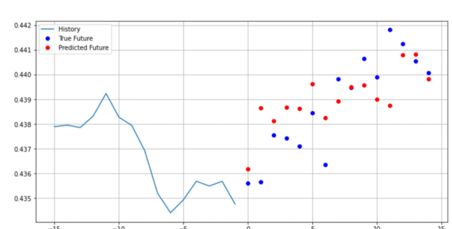

Fig. 7: *represents a random sample for multivariate time series multistep forecasting using GRU*

The result from the Table. V looks so promising, as can be seen the error for Univariate single step the error is ± 1% in either direction,

		Univariate		Multivariate	
Dataset	Model	MSE	MAPE	MSE	MAPE
Train set	Decision Trees	0.000465	2.997	0.000452	2.978
Test set		0.000680	3.707	0.000679	3.703
Train set	Linear Regression	0.000448	2.967	0.000450	2.980
Test set		0.000654	3.650	0.000738	3.973
Train set	Random Forest	0.000687	1.847	0.000226	2.021
Test set		0.000198	3.758	0.000737	4.014
Train set	Support vector regressor	0.000453	2.961	0.000505	3.345
Test set		0.000661	3.654	0.00585	13.09
Train set	XGB Regressor	0.000385	2.880	0.000638	4.403
Test set		0.000667	3.683	0.001058	5.56

TABLE III: *15 min resampling data results for univariate and multivariate time series Forecasting*

		Univariate		Multivariate	
Dataset	Model	MSE	MAPE	MSE	MAPE
Train set	Decision Trees	0.00248	6.139	0.00197	5.710
Test set		0.00595	9.273	0.00582	9.084
Train set	Linear Regression	0.00247	6.124	0.00209	5.864
Test set		0.00595	9.265	0.00148	16.053
Train set	Random Forest	0.00162	4.793	0.001144	4.141
Test set		0.00614	9.450	0.00595	9.544
Train set	Support vector regressor	0.00235	6.032	0.00721	6.875
Test set		0.00724	10.018	0.00257	11.057
Train set	XGB Regressor	0.0078	5.746	0.00726	4.802
Test set		0.01	9.441	0.01	9.697

TABLE IV: *1 Day resampling data results for univariate and multivariate time series Forecasting*

		Univariate single step forecasting		Multivariate single step forecasting		Multivariate multi step forecasting	
Dataset	Model	MSE	MAPE	MSE	MAPE	MSE	MAPE
Train set	LSTM	0.0000388	0.9152	0.0000484	1.053	0.0000468	3.2915
Test set		0.0000451	1.0007	0.000774	3.836	0.000249	2.5803
Train set	GRU	0.0000368	0.899	0.000093	1.025	0.0000487	3.2748
Test set		0.0000417	0.968	0.00001	1.203	0.00054	3.1007

TABLE V: *univariate and multivariate time series Forecasting using LSTM and GRU*

15 min Univariate	15 min Multivariate
DT, LR, SVR and XGBoost	DT and LR

TABLE VI: *Best ML Models for forecasting 15 min*

1D Univariate	for 1D Multivariate
DT, LR	DT

TABLE VII: *Best ML Models for forecasting for 1D.*

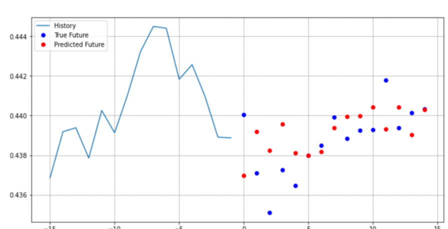

Fig. 8: *represents a random sample for multivariate time series multistep forecasting using GRU*

which is good for this setting. For multivariate single step forecasting GRU perform better than LSTM, there is some slight overfitting problem with LSTM. While for multivariate multi step forecasting LSTM are performing well by a marginally small value, nevertheless both of them can be used for this process of forecasting.

So based on all the results we can argue empirically that DL models performed better task in forecasting the price of Ethereum

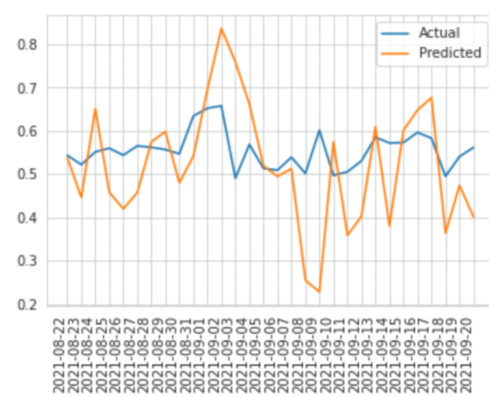

Fig. 9: *Multivariate Forecasting with Linear regression for 15min.*

even with the presence of outliers and is not overfitting to the data as well.

Further analysis showed that 8 out of 20 times LSTM can predict the trend as well as the price very close to the actual price and GRU 22 out of 50 time could predict the trend when randomly sampled from the dataset.See, Fig.7 and Fig.8.

VII. Conclusion

This research project leveraged all the models from traditional statistical models to the state-of-the-art models to predict time series forecasting task that is there in the literature. This paper compared all

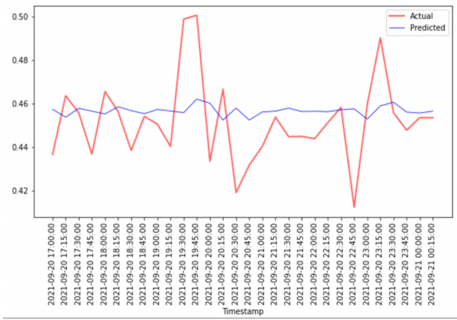

Fig. 10: *Univariate Forecasting with SVR for 15min*

the models using the metric as MAPE. Our findings showed that DL models LSTM and GRU outperforms various other models in terms of univariate and as well multivariate forecasting tasks. And also, the price prediction is close to the actual price sometimes but not always, even when there are outliers in the data, and hence LSTMs and GRUs can be used for the forecasting the price of Ethereum. By resampling the data, the error in forecast actually increased. So, resampling the data can never help in actual price prediction.

The univariate for 15 min forecasting performed better compared to the multivariate forecasting. But, for 1 day multivariate and univariate forecasting showed that there was slight overfitting problem even after hyperparameter tuning. Intuitively speaking, as univariate forecasting is single step forecast so the error is based on only single timestep. Hence, the error will be small. In the case of Multi step forecasting the error is added for the next timesteps and this will give out a error a bit higher than univariate forecasting.

But the actual price forecasting still remains questionable, maybe we can come up with different loss function which can punish these models when the forecast is wrong, as well as different hybrid models for predicting actual price of Ethereum. This model doesn't consider the sentiment of the crypto market. We can combine sentiments of the market in combination with the technical indicators.

Instead of posing it as a regression task maybe we can perform the classification task by restructuring the data and can do 3 class (up, down, and neutral) classification for the trend forecasting using the various models. So that performance of these models can do a lot better than a regression task.

References

[1] R. Hyndman and G. Athanasopoulos, "Forecasting: principles and practice, 2nd edition," *OTexts*, 2018.

[2] "Sentiment analysis on social media for stock movement prediction," *Expert Systems with Applications*, vol. 42, no. 24, pp. 9603–9611, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417415005126

[3] B. G. Malkiel, *A random walk down Wall Street : the time-tested strategy for successful investing*. Completely revised and updated edition. New York : W.W. Norton, [2003]. [Online]. Available: https://search.library.wisc.edu/catalog/999956933302121

[4] B. Malkiel, "A random walk down wall street: The time-tested strategy for successful investing," 01 2007.

[5] A. W. Lo and A. C. MacKinlay, "Stock Market Prices Do Not Follow Random Walks: Evidence from a Simple Specification Test," *The Review of Financial Studies*, vol. 1, no. 1, pp. 41–66, 04 2015.

[6] A. Ang and G. Bekaert, "Stock Return Predictability: Is it There?" *The Review of Financial Studies*, vol. 20, no. 3, pp. 651–707, 07 2006.

[7] M. Obthong, N. Tantisantiwong, W. Jeamwatthanachai, and G. Wills, "A survey on machine learning for stock price prediction: algorithms and techniques," May 2020, pp. 63–71. [Online]. Available: https://eprints.soton.ac.uk/437785/

[8] D. W. Gareth James and R. T. Trevor Hastie, "An introduction to statistical learning : with applications in r," [2013], includes bibliographical references and index. [Online]. Available: https://search.library.wisc.edu/catalog/9910207152902121

[9] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207006000239

[10] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015. [Online]. Available: https://arxiv.org/abs/1506.00019

[11] "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3234–3241, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417414007684

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[13] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks:: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998. [Online]. Available: https://EconPapers.repec.org/RePEc:eee:intfor:v:14:y:1998:i:1:p:35-62

[14] A. Adebiyi, A. Adewumi, and C. Ayo, "Comparison of arima and artificial neural networks models for stock price prediction," *Journal of Applied Mathematics*, vol. 2014, pp. 1–7, 03 2014.

[15] R. Ahangar, M. Yahyazadehfar, and P. Hassan, "The comparison of methods artificial neural network with linear regression using specific variables for prediction stock price in tehran stock exchange," *International Journal of Computer Science and Information Security*, vol. 7, 03 2010.

[16] "Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network," *Applied Soft Computing*, vol. 58, pp. 35–52, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494617301862

[17] K. Lin, Q. Lin, C. Zhou, and J. Yao, "Time series prediction based on linear regression and svr," in *Third International Conference on Natural Computation (ICNC 2007)*, vol. 1, 2007, pp. 688–691.

[18] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, Dec 1989.

[19] J. B. Singh, "Current approaches in neural network modeling of financial time series," 2011.

[20] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning : A systematic literature review: 2005–2019," *Applied Soft Computing*, vol. 90, p. 106181, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620301216

[21] M. Rahman, M. Hossain, T.-S. Junaid, M. Forhad, and M. Hossen, "Predicting prices of stock market using gated recurrent units (grus) neural networks," vol. 19, pp. 213–222, 01 2019.

[22] H. El, A. Mohamed, and H. Fawzy, "Time series forecasting using tree based methods," p. 229, 01 2021.

[23] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1643–1647.

[24] J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters," *Procedia Computer Science*, vol. 125, pp. 676–682, 2018.

[25] "https://www.kaggle.com/competitions/g-research-crypto-forecasting/data." [Online]. Available: https://www.kaggle.com/competitions/g-research-crypto-forecasting/data

[26] "https://www.investopedia.com/terms/m/movingaverage.asp." [Online]. Available: https://www.investopedia.com/terms/m/movingaverage.asp

[27] "https://www.investopedia.com/terms/b/bollingerbands.asp." [Online]. Available: https://www.investopedia.com/terms/b/bollingerbands.asp

[28] "https://www.investopedia.com/terms/m/macd.asp." [Online]. Available: https://www.investopedia.com/terms/m/macd.asp

[29] "https://www.investopedia.com/terms/r/rsi.asp." [Online]. Available: https://www.investopedia.com/terms/r/rsi.asp

[30] "https://www.investopedia.com/terms/v/vwap.asp." [Online]. Available: https://www.investopedia.com/terms/v/vwap.asp

[31] P. Samuels and M. Gilchrist, "Pearson correlation," 04 2014.

MSc Project - Reflective Essay

Project Title:	Univariate and Multivariate Time Series Forecasting in Ethereum
Student Name:	Ramakrishna Puttala Karun Kumar
Student Number:	2120542309
Supervisor Name:	Dr. Emmanouil Benetos
Programme of Study:	MSc Big Data Science

The research project is said to primarily focus on the problem of the forecasting the highly volatile market of the cryptocurrency (like Ethereum) using Machine learning and Deep learning models. As the financial sector has been quite recently introduced to the term cryptocurrency and its trading, there has been a high priority to develop and assess a model that can give best future predictions by keeping up with the said volatility of the market. Here, the steps taken for analysis of their impact and improving the forecasting such as resampling the data into different time frames, comparing Uni-variate and multi-variate using the technical indicators by the aid of ML and DL models. The essay provides the overview of the challenges faced and solved with the defined approach to solve the forecasting and addressing the effect of these challenges on the predictions due to the errors that are analysed. Furthermore, a future work on this approach is proposed to further overcome these limitations.

Strengths

- Designed a good feature extraction stage within a ML pipeline that is usually used by traders and investors that helped the model to predict the target accurately with the DL models.
- The trained machine learning and deep learning models from the project are able to predict the patterns reasonably well.
- With the LSTM/GRU, inference drawn was that more than 40% of time the model is able to predict the target trend, which shows that if we follow the proper risk management of 1:2 risk to reward we can earn up to 20% return on investment every month, but the samples were drawn at random.
- DL models by far outperform even with the presence of the outliers.

Weakness

- Actual target prediction is still debateable with these model even with the state-of-the-art DL models.
- The Machine learning model performance dropped compared to the DL models with the presence of the outliers.
- Some ML models are only able to predict the linear trend of a crypto's development and this concludes that the historical OHLCV data of the crypto asset contains only the explanatory value for a linear trend prediction.
- The actual data had missing values in the target features and due to spline interpolation technique for handling the missing data contributed to the irreducible error in the model.
- The loss function decreases quickly at first, and it suffers from the occasional value explosions and vanishes this is because of the loss function.
- Resampling the data into 1D reduced the data set from 1.9 million to about 1.5K datapoints, but this resulted in the increase in forecasting error due to information lost while resampling.

Presentation of possibilities for further work

In this project, only the Ethereum dataset was used but we can consider other crypto currencies dataset and perform analysis on them. We can see if there is some price correlation between other crypto assets, and perform analysis in the price prediction, a small increase in predictive efficiency is always profitable.

In addition to historical prices, other related information such as politics, economic growth, financial news, and social media mood could have an impact on stock prices. Many studies have shown that sentiment (mood) analysis has a significant impact on future prices. As a result, combining technical and fundamental analyses can yield a highly efficient prediction (Mehtabhorn Obthong et al., 2020). In order to overcome the actual target prediction, we can pose this problem as a trend forecasting i.e., a classification task instead of a regression task. And perform the analysis using the models discussed in the paper.

Further, develop algorithmic trading on the trend prediction using the ML and DL models, and can perform portfolio optimisation techniques based on the pairs of the asset groups on these cryptocurrencies for best returns.

In order to increase the performance of the model, hybrid models like combing ML can be considered for the forecasting process. LSTM and its forms along with other hybrid models dominate the financial time series forecasting (Omer Berat Sezer et al., 2020).

With more time in project, I would have implemented time series forecasting using the state-of-the-art Transformer model using the time embedding. In the recent times transformers have gained a lot of popularity due to their great performance. Transformers bring the self-attention mechanism, parallelization, and positional encoding under one roof which provides an edge over classical LSTMs. Instead of positional encoding the transformer model for time series, time embedding operation should be performed. Since a Transformer requires a notion of time when processing the target price of the Ethereum, without time-embedding the Transformer model wouldn't receive any information about the chronological order of our target price. Time embedding is performed by Time2Vec (Seyed Mehran Kazemi et al., 2019) describes about the time component which is comprised of both periodic and non-periodic patterns. The time vector is combined with the input sequence and passed to transformer architecture (Ashish Vaswani et al., 2017) with hyperparameter tuning for good forecasting results.

Critical analysis of the relationship between theory and practical work produced

As a quite experienced trader in the stock market and cryptocurrency's since past two years, I have been curious regarding the systems and techniques that mitigate the risks and increase the profits from each trade. The result of the knowledge which had to amalgamate both the theory and skills that I learnt during the trading and the skills that I acquired during the course to predict the stock prices respectively. The strong inclination towards the working for the stock market motivated the idea of doing a project in the field of finance. The path led me to pursue Masters in Big Data Science with the primary motivation of enhancing my practical knowledge in field of Data and Machine Learning. As, the volume of stock market has been continuously rising exponentially, no ordinary human can handle, analyse, capture most of the financial market data in the small span of time for intra-day trading. The automated analysis of financial market motivated to develop a Machine learning financial cryptocurrency forecasting project with most of the skills that were learnt during the course. My innate financial literacy pushed me to successfully forecast the trend of the crypto market of Ethereum with various ML models. And, making highly affecting decisions, modification to the current models' implementations to further study, understand, and enhance the forecasting while reducing the latency and improving the performance.

At the beginning stages of building these models, I struggled to discover a new approach with the pre-processing of data, and this is because every business problem has different approach and constraints. So, in the literature for time series forecasting there were many different kinds of solutions for handling the missing values and is not straight forward. Spline interpolation was the best approach for the problem that I had.

With respect to the model buildings, in theory, The ARIMA models only work with the linear patterns in the data and cannot capture the non-linear behaviours and experiments in our project supported this statement. And ML models like SVR and XGBoost, Trees based algorithms from the literature exhibited that they are capable of learning the non-linear behaviour in the data and empirical results are in accordance with the theory. The Deep learning models are built to learn the complex and non-linear behaviour in the data. In LSTMs and GRUs the weights are learnt using the back propagation through time. LSTMs deep Learning models are widely preferred for time series forecasting. DL models beat ML models in terms of predictions, but can easily overfit to the data. The experiment performed in the project dissertation proved that DL models are the best in terms of the predicting the target.

Awareness of Legal, Social, Ethical Issue and Sustainability

The data is obtained from Kaggle, which is an open-source public data platform available for research in various domains in data science. Data is collected from the host who collects it and hosts a competition for a prize money on the Kaggle platform. The raw OHLCV data along with the other features from the train set are publicly available for all the traders and investors across the world and hence this data is transparent and can be used by anyone for research and self-study related purpose. Hence it doesn't violate any legal or ethical grounds.

Nothing in this project should be construed as a recommendation that a specific asset, portfolio of securities, transaction, or investing strategy is appropriate for any particular individual or an organisation. Trading in financial instruments especially like the crypto currencies carries a high risk of loss and is not recommended for all investors with regard to their risk appetite. Due to possible fluctuations in these assets' clients run the risk of losing more money than they initially invested. As the data set is not complete and had to come up with strategies to fill the gaps in the data and this updated dataset doesn't resonate the original data. With this reason errors in the predictions can be costly and hence not recommended.

Training a deep learning requires a lot of computation that's the reason why these models run on GPUs and not the CPUs most of the times. So, training these models on the GPUs for the datasets which are quite big, and complex might leave a high-level carbon-footprint. But these models were trained on the college GPU servers which are sustainable to the environment.

Conclusion

Overall, this project gave me a sense of satisfaction after training and testing the models. Comparing the theory and practical approaches made me realise what these models are capable of and its weakness. Coming from a no programming background and as a trader, I'm really proud of the work that I had progressed during this project. And next step is taking the insights from this project to build an algorithmic trading model which

I’m very excited about and start my own venture. I am positive that this project is going to help me succeed in my professional ventures.

I would like to extend my gratitude to my supervisor, Dr. Emmanouil Benetos for supporting and guiding me towards the right path in the project that I had chosen right from the beginning.

References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need- 2017.

Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, Marcus Brubaker. Time2Vec: Learning a Vector Representation of Time-2019.

Mehtabhorn Obthong, Nongnuch Tantisantiwong, Watthanasak Jeamwatthanachai, and Gary Wills.A Survey on Machine Learning for Stock Price Prediction: Algorithms and Techniques-2020.

Omer Berat Sezer,Mehmet Ugur Gudelek,Ahmet Murat Ozbayoglu. Financial time series Forecasting with deep learning: A systematic literature review: 2005-2019.