Predicting Cryptocurrency Prices Based On Time Series Analysis And Machine Learning

1st Nguyen Cam Tu
IS403.N22.HTCL
University of Information Technology
Ho Chi Minh city
20520837@gm.uit.edu.vn

2nd Nguyen Thi Thu Thuy
IS403.N22.HTCL
University of Information Technology
Ho Chi Minh city
20520797@gm.uit.edu.vn

3rd Nguyen Thanh Son
IS403.N22.HTCL
University of Information Technology
Ho Chi Minh city
20521847@gm.uit.edu.vn

***Abstract*—This study presents a framework for cryptocurrency price prediction using a combination of nine machine learning algorithms: RNN, LSTM, DNN, ARIMA, ARIMAX, time series clustering, KNN, and time series anomaly detection. Historical price data and technical indicators are utilized as input features to train and evaluate the prediction models. The performance of each algorithm is compared and analyzed using real-world cryptocurrency datasets. The findings demonstrate the effectiveness of these algorithms in predicting cryptocurrency prices, providing valuable insights for investors in navigating the volatile cryptocurrency markets. However, it is important to exercise caution and implement appropriate risk management strategies when making investment decisions based on these predictions.**

***Keywords*—Time Series Forcasting, RNN, LSTM, DNN, KNN, Linear Regression, ARIMA, ARIMAX, Time Series Clus- tering, Time Series Anomaly Detection, K-Means.**

I. INTRODUCTION

Cryptocurrencies have gained widespread popularity, attracting both individual and institutional investors. However, the volatile nature of cryptocurrency markets poses challenges for investors seeking to make informed decisions. Accurate prediction of cryptocurrency prices is crucial for maximizing profits and minimizing risks. This study aims to develop a framework using time series analysis and machine learning algorithms to predict cryptocurrency prices. By leveraging advanced techniques such as RNN, LSTM, DNN, ARIMA, ARIMAX, time series clustering, KNN, and time series anomaly detection, we seek to provide valuable insights for investors navigating the dynamic cryptocurrency markets.

II. RELATED WORKS

Yongqiong Zhu [1] used RNNs to predict stocks price and admit that RNN is very suitable for predicting stocks. Gabor Petnehazi [2] used RNN Network and LSTM Network for time series forcasting. The objective of this study was to investigate and elucidate various facets of employing RNNs for the purpose of time series forecasting. This idea is significant as it allows us to build neural networks that can process time-dependent values. Benjamin Lindemann, Timo Müller, Hannes Vietz,Nasser Jazdi, Michael Weyrich [3] did a survey on long short-term memory networks for time series prediction . They give an overview of LSTM architectures that are developed to predict nonlinear time series behavior.

Aysun Bozantaa, Sean Berrya, Mucahit Cevika, Beste Bulutb, Deniz Yigitb, Fahrettin F. Gonenb, and Ayse Basara [4] investigated alternative clustering methodologies to group the products based on the price patterns and sales volumes. They found that our proposed clustering approach and image clustering both perform well for finding the products with similar price and sales patterns within large datasets. Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah [5] used the K-Means clustering algorithm to partition a set of unlabelled objects into k clusters, ensuring that each cluster contains at least one object. Ali Javed, Byung Suk Lee, and Donna M Rizzo [6] conducted a benchmark time series clustering study using the UCR archive datasets. They combined k-means and density-based clustering algorithms with Euclidean and DTW distances. Francisco Martinez Alvarez, Alicia Troncoso, Jose C. Riquelme, and Jesus S. Aguilar Ruiz [7] used K-means clustering to group and label time series data on electricity demand. Before performing time series forecasting, they employed K-means clustering to group and label the time series data. One of the measures they used to evaluate the performance of the clustering algorithm is by using the silhouette index. Afterwards, they proceeded to predict electricity prices of Spanish, Australian, and New York markets using various forecasting algorithms such as ARIMA, KNN, SVM. . . .

Yuhan [8] conducted research on predicting stock prices using multiple regression models. The paper focuses on applying the multiple regression method to identify and measure the relationship between the dependent variable (stock price) and multiple independent variables (financial indicators, market data, and other relevant factors).

Mojtaba Nabipour, Pooyan Nayyeri, Hamed Jabani, and Amir Mosavi [9] focus on the application of deep learning in stock market prediction. They explain how deep learning is used to forecast stock prices and market fluctuations. The paper provides an overview of popular deep learning architectures and discusses the advantages and limitations of deep learning. It also addresses potential directions for future development in this field.

Cyril Bachelard, Apostolos Chalkis, Vissarion Fisikopoulos, and Elias Tsigaridas from the Faculty of Business and Economics (HEC) at the University of Lausanne, Switzerland,

and the National and Kapodistrian University of Athens, Greece, along with GeomScale.org, Inria Paris, and IMJ-PRG at Sorbonne Universite and Paris Universite [10] focus on applying randomized geometric tools for anomaly detection in the stock market. These tools are used to analyze market data and identify unusual or outlier patterns. The paper introduces specific methods and algorithms, emphasizing the significant differences and deviations from conventional models. It also examines the effectiveness and advantages of randomized geometric methods and discusses potential directions for future development in this field.

Matthew Chen Neha Narwal and Mila Schultz [11] acknowledged the best performance was achieved by the Automated Regression Integrated Moving Average (ARIMA) model, attributed to its features and relevance with time series data with accuracy 61.17.

Richard M. N. Y. Sarpong-Streetor, Rajalingam Sokkalingam , Mahmod Othman , Hanita Daud, and Derrick Asamoah Owusu [12] has examined the ability of ARIMAX to model the fuel price of Ron97 using times series data of Ron97 and another exogenous time series, the crude oil price, in this case OPEC in Malaysia. It is possible to do the modelling and forecasting accurately using the ARIMAX. The findings demonstrate that incorporating external variables improves the accuracy of the ARIMAX model compared to ARIMA.

Saad Ali Alahmari [13] compared and assesses tow machine-learning methods (DTR and KNN) for predicting cryptocurrency using the datasets of three-pricemajor cryptocurrencies: Bitcoin, XRP and Ethereum. The dependent variable is time series continuous and predictor variables are all continuous. Many studies have been performed in the past experimental, however, very few compared more than one model such as DRT and KNN with several cryptocurrencies. It is remarkable that the Decision Tree Regressor (DTR) outperforms the K-Nearest Neighbor (KNN) in terms of MAE, MSE, RMSE.

III. MODELING

A. Data Preparation

We use 3 dataset files TRX-USDT, DOGE-USD and XLM-USD Start date: 01/12/2017 End date: 10/06/2023

B. RNN

The origin of RNN (Recurrent Neural Network) can be traced back to the research efforts of scientists in the field of artificial intelligence and machine learning. In the neural network revolution, Paul Werbos made a significant contribution to the development of RNN through his research work in 1974. In his doctoral thesis, Werbos introduced the concept of recurrent neural networks and the backpropagation algorithm for training such networks.

However, the practical interest and development of RNN gained momentum after the publication of a research paper by Ronald J. Williams and David Zipser in 1989. In their paper titled ”A Learning Algorithm for Continually Running Fully Recurrent Neural Networks,” Williams and Zipser introduced the concept of reinforcement learning for RNN, allowing the neural network to maintain hidden states and learn through continuous interaction with the environment.

C. LSTM

LSTM (Long Short-Term Memory) is a recurrent neural network (RNN) architecture introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997. LSTM was developed to address the ”vanishing gradient” problem of traditional RNNs, where the gradient approaches zero during backpropagation. This makes it difficult to retain and store information over long sequences.

LSTM solves this issue by using gated mechanisms to regulate the process of information storage and retrieval within the network. Specifically, LSTM employs three main gates: the forget gate, the input gate, and the output gate. These gates enable LSTM to control the flow of information through time steps.

In 1997, Hochreiter and Schmidhuber published the paper ”Long Short-Term Memory”, which presented the LSTM architecture and its functioning. This paper opened up a new research direction in the field of recurrent neural networks and became a significant foundation for the development of LSTM in practical applications.

D. DNN

DNN (Deep Neural Network) originates from the theoretical foundation of reinforcement learning and multilayer neural networks in research from the 1940s and 1950s. However, the true development of DNN has occurred in recent years, thanks to advancements in computing technology and parallel computing capabilities, such as GPU clusters, as well as the availability of large and powerful datasets for model training.

DNN operates by utilizing a deep neural network with multiple hidden layers to learn and extract information from input data. Each hidden layer in DNN consists of a number of interconnected neurons with associated weights. These weights are adjusted during the training process so that the model can learn and represent complex patterns and features of the data.

E. ARIMA

Autoregressive integrated moving average (ARIMA) models are models which can be fitted to a single time series and used to make predictions of future observations. They owe their popularity primarily to the work of Box and Jenkins (1970), who defined the class of ARIMA and seasonal ARIMA models and provided a methodology for selecting a suitable model from that class.

An ARIMA model is composed of 3 constituent units which are :

- AR: Autoregression. This part explores any dependent relationship between an observation and some number of lagged variables.
- I: Integrated. This part aims to make time-series stationery by subtracting or differencing an observation from

observation at the previous time step of the same time series.

- MA: Moving Average. This part explores the relationship between an observation and a residual error by application of moving average to lagged observations, with any given time window.

F. ARIMAX

ARIMAX is the machine learning model used for time series analysis. It takes data and make observations. On the basis of previous observations it takes mean average and differentiate between consecutive two time-stamps in order to make time series stationary. The mathematical expression for the ARIMAX model is given below:

$$\Delta \boldsymbol{Y}_t = \varepsilon_t + \sum_{i=1}^{p} \varphi_i \, \Delta \boldsymbol{Y}_{t-i} \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \sum_{m=1}^{M} \boldsymbol{\beta}_m \boldsymbol{X}_{t-\mathrm{m}}$$

G. KNN

The k-nearest neighbors (KNN) algorithm was first introduced in 1951 by Fix and Hodges for non-parametric discriminant analysis. Since then, it has been widely used in pattern recognition and machine learning applications.

H. Linear Regression

Linear regression algorithm has long existed in the field of statistics and regression. It is one of the simplest and fundamental algorithms in machine learning and has been developed since the early 19th century.

The linear regression algorithm works by finding the best line (or hyperplane) to model the linear relationship between input variables and output values. It uses the method of least squares, where the parameters are adjusted to minimize the error between the predicted values and the actual values.

I. Time Series Clustering

The origin of time series clustering can be traced back to various fields and sources. Here are some important origins of time series clustering:

Time series clustering has origins in multiple fields such as statistics, macrostatistics, artificial intelligence, data mining, and real-world applications. The combination of these origins has led to the development of modern methods and techniques for clustering time series.

J. Time Series Anomaly Detection

The Time series anomaly detection algorithm has existed for a long time in the field of data analysis and forecasting. However, with the advancement of technology and the need to detect anomalous events in time series data, this algorithm has become increasingly popular and studied more deeply.

The Time series anomaly detection algorithm works by identifying data points in a time series that do not follow the usual pattern or behavior. To do this, the algorithm uses methods such as statistical analysis, machine learning, and neural networks. Models are built based on historical data to estimate patterns and model the normal variations in the time series. When new data is available, the algorithm compares it to the learned model to determine if the data is anomalous or not. The Time series anomaly detection algorithm plays an important role in detecting anomalies in time series data, improving operational processes, safeguarding data, and conducting data analysis research.

IV. METHODOLOGY

A. Single Model

1) RNN: RNN is a deep learning network structure. It is designed to handle sequence data. The key feature of an RNN is its ability to consider the sequential nature of the input data by introducing loops within the network. This loop allows information to be passed from one step to the next, which enables the network to capture dependencies and patterns in the data over time. In time series analysis, RNNs can be trained to predict future values in a time series based on past observations. By using the historical data, the RNN can learn patterns and trends in the time series, allowing it to make predictions.

2) LSTM: Long Short-Term Memory (LSTM) is based on the idea of gates. The main element of LSTM is the cell state (Ct) that can “remember” long-term dependencies, and the gates decide how the information flow.

The input gate (1) decides how much newly received information is added to the current cell state (4). Additionally, this gate applies appropriate transformations to the newly received information.

The forget gate (2) decides which part of the cell state from the previous period will be forwarded. This gate receives a combined vector consisting of the hidden state (5) from the previous period and the input from the current period . Forget gate returns a value from 0 to 1, where 0 means complete “forgetting” and 1 means all the information from the previous cell state is transferred.

Finally, the output flow is controlled by the output gate (3). It first decides what parts of the cell state are output and then transforms them accordingly using the tanh function.

$$i_t = sigmoid\,(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = sigmoid\,(W_f x_t + U_i h_{t-1} + b_f) \quad (2)$$

$$o_t = sigmoid\,(W_o x_t + U_i h_{t-1} + b_o) \quad (3)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$h_t = o_t \cdot tanh(\, c_t) \quad (5)$$

LSTM neural network is a powerful tool in the field of machine learning. It can extract features, dimensions, and improve data classification. It is a machine language with the ability to learn internal representation and solve complex combinatorial problems.

3) DNN: The Deep Neural Network (DNN) model is a multilayer machine learning model used in artificial intelligence. It is based on the structure of artificial neural networks, with multiple hidden layers between the input layer and the output layer. DNN is renowned as a deep neural network because it has the ability to learn and understand complex features from input data.

4) ARIMA: Autoregressive integrated moving average (ARIMA) models are models which can be fitted to a single time series and used to make predictions of future observations. They owe their popularity primarily to the work of Box and Jenkins - 1970, who defined the class of ARIMA and seasonal ARIMA models and provided a methodology for selecting a suitable model from that class.

An ARIMA model is composed of 3 constituent units which are :

- AR: Autoregression. This part explores any dependent relationship between an observation and some number of lagged variables.
- I: Integrated. This part aims to make time-series stationery by subtracting or differencing an observation from observation at the previous time step of the same time series.
- MA: Moving Average. This part explores the relationship between an observation and a residual error by application of moving average to lagged observations, with any given time window.

5) ARIMAX: ARIMAX stands for AutoRegressive Integrated Moving Average with eXogenous variables. The name ARIMAX is as extension of the ARIMA respectively. The X added to the end stands for "exogenous". In other words, it suggests adding a separate different outside variable to help measure our endogenous variable.

An exogenous variable is a variable that is not affected by other variables in a model but can affect the dependent variable. Some examples of exogenous variables include income tax rates, weather, the presence of pests, and interest rates. These variables are considered exogenous because they are not influenced by other variables in the model. For example, the amount of rainfall cannot be affected by the amount of fertilizer used or the type of soil used in a crop yield model.

6) Linear Regression: Linear regression, a method of machine learning, the model is based on a pair of two variables, the independent variable (x) and the dependent variable (y). In the context of multiple linear regression, there may be many independent variables. A simple linear regression has only one independent variable x. In the conditions given by the current model, ... the data set, there is only one independent variable, date. The first date that rises to the length of the date vector is represented by the integer . The length of this vector must be an integer and the date variable will change according to the time and, at the same time, the price of the stock, which is the dependent variable, will also change.

B. Hybrid Model

1) Hybrid model based on TIME SERIES CLUSTERING with RNN, LSTM, DNN: The K-Means clustering algorithm was use, along with two commonly used distance metrics: Euclidean distance and Dynamic Time Warping (DTW) distance. These metrics assessed the similarity between time series data, and k-means clustering was applied to cluster the data accordingly.

The performance of the generated clusters was evaluated using the silhouette score, which measures the cohesion within clusters and the separation between clusters. Higher silhouette scores indicate well-defined and well-separated clusters, while lower scores suggest overlapping or poorly separated clusters. By applying k-means clustering with different distance metrics and evaluating the results using the silhouette score, the aim was to determine the most suitable clustering approach for analyzing cryptocurrency closing prices.

Subsequently, the study focused on predicting the closing prices of Bitcoin using advanced algorithms such as RNN, DNN, and LSTM.

Step 1: Apply KMeans

- Create dataset with time step = 100.
- Using TimeSeriesScalerMeanVariance() to scaled data.
- Using sum of squared distances to decide cluster number.
- Apply Kmeans using euclidean distance.
- Apply Kmeans using dynamic time warping (dtw) distance.
- Divide dataset into 3 clusters and start to predict.

Step 2: Apply RNN, LSTM, DNN TO predict clusters.

Step 3: Evaluate RMSE and MAE.

Step 4: Choose the best model to predict each ratio.

2) Time Series Anomaly Detection: The Time series anomaly detection algorithm has existed for a long time in the field of data analysis and forecasting. However, with the advancement of technology and the need to detect anomalous events in time series data, this algorithm has become increasingly popular and studied more deeply.

The Time series anomaly detection algorithm works by identifying data points in a time series that do not follow the usual pattern or behavior. To do this, the algorithm uses methods such as statistical analysis, machine learning, and neural networks. Models are built based on historical data to estimate patterns and model the normal variations in the time series. When new data is available, the algorithm compares it to the learned model to determine if the data is anomalous or not. The Time series anomaly detection algorithm plays an important role in detecting anomalies in time series data, improving operational processes, safeguarding data, and conducting data analysis research.

V. RESULT

A. Materials

TABLE I
DESCRIPTIVE STATISTIC TABLE

Measure	**TRX-USD**	**DOGE-USD**	**XLM-USD**
Mean	0.047120	0.062898	0.184427
Std	0.029368	0.094856	0.133435
Min	0.002073	0.001537	0.033441
Max	0.220555	0.684777	0.896227

B. Evaluation

- MAE: MAE measures the average magnitude of the errors made by a model in predicting the target variable. It is calculated by taking the average of the absolute differences between the predicted values and the actual values. MAE is useful because it gives equal weight to all errors without considering their direction (positive or negative).

$$MAE = \overset{\text{Divide by the total number of data points}}{\boxed{\frac{1}{n}}} \underset{\text{Sum of}}{\sum} \underbrace{\left| \overset{\text{Actual output value}}{\boxed{y}} - \overset{\text{Predicted output value}}{\boxed{\hat{y}}} \right|}_{\text{The absolute value of the residual}}$$

- RMSE: RMSE is another commonly used metric that measures the square root of the average of the squared differences between the predicted values and the actual values. RMSE penalizes larger errors more heavily than MAE since it squares the errors before taking the average.
- MAPE: MAPE measures the average percentage difference between the predicted and actual values. It is commonly used when the relative error between the predicted and actual values is more important than the absolute

$$MAPE = \frac{\overset{\text{Multiplying by 100\% converts to percentage}}{\boxed{100\%}}}{n} \sum \left| \frac{\overbrace{y - \hat{y}}^{\text{The residual}}}{\underset{\text{Each residual is scaled against the actual value}}{\boxed{y}}} \right|$$

error.

C. Result

TABLE II
TRX-USD VALIDATION

Model	Ratio	MAE	RMSE	MAPE
RNN	7-2-1	0.235	0.236	77.05
	5-3-2	0.217	0.219	77.2
	6-2-2	0.216	0.218	76.9
LSTM	7-2-1	0.236	0.237	77.32
	5-3-2	0.216	0.219	77.11
	6-2-2	0.032	0.039	26.62
DNN	7-2-1	0.002	0.002	2.547
	5-3-2	0.002	0.003	3.494
	6-2-2	0.002	0.003	3.868
ARIMA	7-2-1	0.033	0.034	54.299
	5-3-2	0.017	0.02	25.642
	6-2-2	0.011	0.013	15.528
ARIMAX	7-2-1	0.033	0.034	54.299
	5-3-2	0.017	0.02	25.642
	6-2-2	0.011	0.013	15.528
KNN	7-2-1	0.03	0.031	48.707
	5-3-2	0.038	0.038	57.167
	6-2-2	0.011	0.013	16.269
LR	7-2-1	0.006	0.007	8.897
	5-3-2	0.089	0.089	136.418
	6-2-2	0.058	0.058	88.746

TABLE III
TRX-USD TEST

Model	Ratio	MAE	RMSE	MAPE
RNN	7-2-1	0.224	0.226	76.81
	5-3-2	0.258	0.288	73.83
	6-2-2	0.271	0.283	76.19
LSTM	7-2-1	0.225	0.227	77.14
	5-3-2	0.258	0.288	73.6
	6-2-2	0.219	0.221	78.2
DNN	7-2-1	0.003	0.004	4.367
	5-3-2	0.006	0.01	7.846
	6-2-2	0.005	0.006	6.165
ARIMA	7-2-1	0.027	0.029	41.365
	5-3-2	0.023	0.032	33.778
	6-2-2	0.028	0.037	29.546
ARIMAX	7-2-1	0.027	0.029	41.365
	5-3-2	0.023	0.032	33.778
	6-2-2	0.028	0.037	29.546
KNN	7-2-1	0.024	0.026	36.38
	5-3-2	0.032	0.04	44.581
	6-2-2	0.024	0.033	24.08
LR	7-2-1	0.016	0.023	20.142
	5-3-2	0.089	0.089	101.871
	6-2-2	0.069	0.069	81.606

TABLE IV
DOGE-USD VALIDATION

Model	Ratio	MAE	RMSE	MAPE
RNN	7-2-1	0.015	0.178	35.16
	5-3-2	0.039	0.045	33.49
	6-2-2	0.038	0.045	32.532
LSTM	7-2-1	0.021	0.024	18.7
	5-3-2	0.026	0.033	21.91
	6-2-2	0.032	0.038	26.62
DNN	7-2-1	0.041	0.041	53.307
	5-3-2	0.005	0.051	64.735
	6-2-2	0.011	0.015	13.021
ARIMA	7-2-1	0.153	0.153	192.462
	5-3-2	0.075	0.076	96.333
	6-2-2	0.027	0.03	32.077
ARIMAX	7-2-1	0.153	0.153	192.462
	5-3-2	0.017	0.02	25.642
	6-2-2	0.027	0.03	32.077
KNN	7-2-1	0.143	0.143	180.44
	5-3-2	0.074	0.076	95.561
	6-2-2	0.021	0.026	25.199
LR	7-2-1	0.133	0.134	168.428
	5-3-2	0.078	0.08	100.996
	6-2-2	0.06	0.062	76.469

TABLE V
DOGE-USD TEST

Model	Ratio	MAE	RMSE	MAPE
RNN	7-2-1	0.048	0.058	35.16
	5-3-2	0.172	0.221	30.49
	6-2-2	0.111	0.136	35.81
LSTM	7-2-1	0.055	0.068	38.68
	5-3-2	0.169	0.218	339.5
	6-2-2	0.108	0.134	34.43
DNN	7-2-1	0.046	0.048	52.762
	5-3-2	0.115	0.141	59.124
	6-2-2	0.057	0.067	28.141
ARIMA	7-2-1	0.114	0.126	137.832
	5-3-2	0.15	0.195	79.616
	6-2-2	0.169	0.196	71.293
ARIMAX	7-2-1	0.114	0.126	137.832
	5-3-2	0.023	0.032	33.778
	6-2-2	0.169	0.196	71.293
KNN	7-2-1	0.106	0.117	128.743
	5-3-2	0.149	0.195	80.675
	6-2-2	0.163	0.19	68.076
LR	7-2-1	0.077	0.09	92.26
	5-3-2	0.152	0.197	88.538
	6-2-2	0.206	0.229	92.234

TABLE VI
XLM-USD VALIDATION

Model	Ratio	MAE	RMSE	MAPE
RNN	7-2-1	0.025	0.278	38.79
	5-3-2	0.023	0.027	35.35
	6-2-2	0.027	0.031	42.01
LSTM	7-2-1	0.025	0.027	38.49
	5-3-2	0.027	0.032	42.7
	6-2-2	0.026	0.03	40.24
DNN	7-2-1	0.004	0.005	4.726
	5-3-2	0.003	0.004	3.689
	6-2-2	0.004	0.006	4.262
ARIMA	7-2-1	0.255	0.255	290.214
	5-3-2	0.024	0.031	21.459
	6-2-2	0.011	0.013	15.528
ARIMAX	7-2-1	0.255	0.255	290.214
	5-3-2	0.017	0.02	25.642
	6-2-2	0.27	0.271	273.43
KNN	7-2-1	0.217	0.217	247.918
	5-3-2	0.015	0.019	14.835
	6-2-2	0.299	0.3	302.112
LR	7-2-1	0.127	0.128	145.49
	5-3-2	0.377	0.378	378.683
	6-2-2	0.108	0.109	107.23

TABLE VII
TRX-USD TEST

Model	Ratio	MAE	RMSE	MAPE
RNN	7-2-1	0.053	0.066	46.806
	5-3-2	0.131	0.169	43.16
	6-2-2	0.082	0.1	30.94
LSTM	7-2-1	0.055	0.069	49.04
	5-3-2	0.136	0.176	46.39
	6-2-2	0.081	0.1	30.88
DNN	7-2-1	0.009	0.01	1 6.021
	5-3-2	0.047	0.072	12.788
	6-2-2	0.021	0.026	7.813
ARIMA	7-2-1	0.16	0.178	120.219
	5-3-2	0.204	0.244	62.529
	6-2-2	0.028	0.037	29.546
ARIMAX	7-2-1	0.16	0.178	120.219
	5-3-2	0.023	0.032	33.778
	6-2-2	0.106	0.128	40.133
KNN	7-2-1	0.131	0.146	98.947
	5-3-2	0.188	0.226	58.926
	6-2-2	0.102	0.124	38.698
LR	7-2-1	0.079	0.091	52.16
	5-3-2	0.391	0.42	142.836
	6-2-2	0.266	0.285	83.139

TABLE VIII
HYBRID MODEL BASED ON TIME SERIES ANOMALY DETECTION WITH ARIMA

Model	Ratio	MAE Validate	RMSE Validate
TRX	7-2-1	0.915	1.173
	5-3-2	0.933	1.193
	6-2-2	0.993	1.27
DOGE	7-2-1	1.032	1.312
	5-3-2	0.959	1.237
	6-2-2	0.89	1.167
XLM	7-2-1	1.241	1.493
	5-3-2	1.019	1.303
	6-2-2	1.195	1.438

TABLE IX
HYBRID MODEL BASED ON TIME SERIES ANOMALY DETECTION WITH ARIMA

Model	Ratio	MAE Test	RMSE Test
TRX	7-2-1	0.915	1.173
	5-3-2	0.933	1.193
	6-2-2	0.993	1.27
DOGE	7-2-1	1.032	1.312
	5-3-2	0.959	1.237
	6-2-2	0.89	1.167
XLM	7-2-1	1.241	1.493
	5-3-2	1.019	1.303
	6-2-2	1.195	1.438

TABLE X

HYBRID MODEL BASED ON TIME SERIES CLUSTERING WITH RNN, LSTM, DNN FOR TRX

Model	Ratio	MAE Validate	RMSE Validate
KMEANS + RNN	7-2-1	0.915	1.173
	5-3-2	0.933	1.193
	6-2-2	0.993	1.27
KMEANS + LSTM	7-2-1	1.032	1.312
	5-3-2	0.959	1.237
	6-2-2	0.89	1.167
KMEANS + DNN	7-2-1	1.241	1.493
	5-3-2	1.019	1.303
	6-2-2	1.195	1.438

TABLE XI

HYBRID MODEL BASED ON TIME SERIES CLUSTERING WITH RNN, LSTM, DNN FOR TRX

Model	Ratio	MAE Test	RMSE Test
KMEANS + RNN	7-2-1	1.006	1.293
	5-3-2	1.009	1.285
	6-2-2	0.96	1.224
KMEANS + LSTM	7-2-1	0.955	1.232
	5-3-2	0.987	1.287
	6-2-2	0.938	1.203
KMEANS + DNN	7-2-1	1.306	1.578
	5-3-2	0.992	1.265
	6-2-2	1.247	1.515

TABLE XII

HYBRID MODEL BASED ON TIME SERIES CLUSTERING WITH RNN, LSTM, DNN FOR DOGE

Model	Ratio	MAE Validate	RMSE Validate
KMEANS + RNN	7-2-1	1.075	1.232
	5-3-2	0.97	1.244
	6-2-2	0.912	1.206
KMEANS + LSTM	7-2-1	0.957	1.213
	5-3-2	0.968	1.271
	6-2-2	0.949	1.236
KMEANS + DNN	7-2-1	1.107	1.43
	5-3-2	0.995	1.268
	6-2-2	1.195	0.004

TABLE XIII

HYBRID MODEL BASED ON TIME SERIES CLUSTERING WITH RNN, LSTM, DNN FOR DOGE

Model	Ratio	MAE Test	RMSE Test
KMEANS + RNN	7-2-1	0.943	1.232
	5-3-2	1.025	1.371
	6-2-2	0.928	1.21
KMEANS + LSTM	7-2-1	0.914	1.225
	5-3-2	0.992	1.29
	6-2-2	0.963	1.294
KMEANS + DNN	7-2-1	1.082	1.382
	5-3-2	1.059	0.008
	6-2-2	1.196	1.466

TABLE XIV

HYBRID MODEL BASED ON TIME SERIES CLUSTERING WITH RNN, LSTM, DNN FOR XLM

Model	Ratio	MAE Validate	RMSE Validate
KMEANS + RNN	7-2-1	0.96	1.215
	5-3-2	0.905	1.139
	6-2-2	0.92	1.2
KMEANS + LSTM	7-2-1	1.048	1.347
	5-3-2	0.954	1.22
	6-2-2	1.028	1.329
KMEANS + DNN	7-2-1	1.201	1.486
	5-3-2	1.032	1.327
	6-2-2	1.09	1.361

TABLE XV

HYBRID MODEL BASED ON TIME SERIES CLUSTERING WITH RNN, LSTM, DNN FOR XLM

Model	Ratio	MAE Test	RMSE Test
KMEANS + RNN	7-2-1	0.91	1.171
	5-3-2	0.96	1.245
	6-2-2	0.924	1.19
KMEANS + LSTM	7-2-1	1.024	1.323
	5-3-2	0.939	1.197
	6-2-2	0.898	1.165
KMEANS + DNN	7-2-1	1.168	1.389
	5-3-2	1.012	1.265
	6-2-2	1	1.235

D. *Visualization*

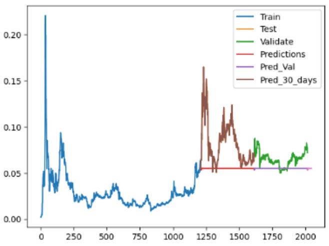

Fig. 1. 30-day prediction of TRX-USD using ARIMA 6:2:2

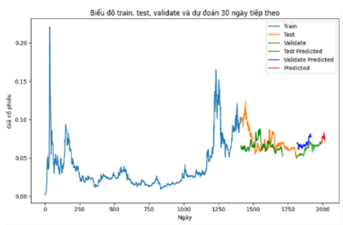

Fig. 2. 30-day prediction of TRX-USD using DNN 7:2:1

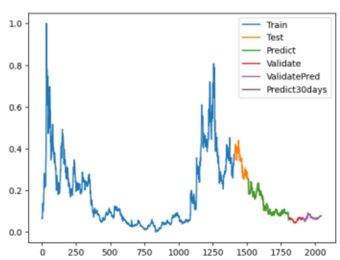

Fig. 3. 30-day prediction of DOGE-USD using RNN 7:2:1

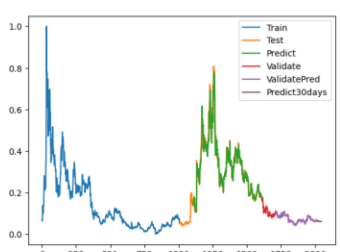

Fig. 4. 30-day prediction of DOGE-USD using RNN 5:3:2

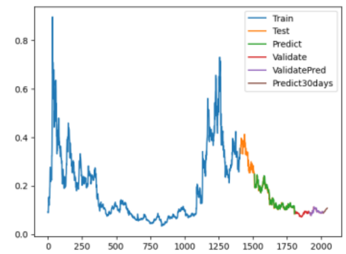

Fig. 5. 30-day prediction of XLM-USD using RNN 6:2:2

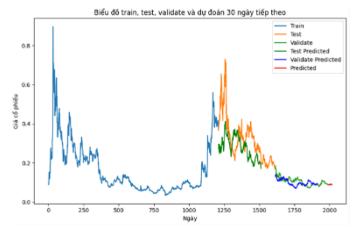

Fig. 6. 30-day prediction of XLM-USD using DNN 7:2:1

VI. CONCLUSION

This study developed a comprehensive framework for predicting cryptocurrency prices using time series analysis and machine learning algorithms. The results demonstrated the effectiveness of the employed algorithms, including RNN, LSTM, DNN, ARIMA, ARIMAX, time series clustering, KNN, and time series anomaly detection, in predicting cryptocurrency prices. The selection of input features and the incorporation of advanced algorithms played a crucial role in improving prediction accuracy. However, the volatile nature of cryptocurrency markets and external factors should be considered when using these predictions for investment decisions. Further research can explore additional data sources and advanced techniques to enhance prediction accuracy. Overall, the framework provides valuable insights for investors, but caution and risk management are essential in cryptocurrency investments.

REFERENCES

[1]Y. Zhu, 'Stock price prediction using the RNN model', J. Phys. Conf. Ser., vol. 1650, no. 3, p. 032103, Oct. 2020, doi: 10.1088/1742-6596/1650/3/032103.

[2]G. Petneh ázi, Recurrent Neural Networks for Time Series Forecasting. 2018.

[3]B. Lindemann, T. M üller, H. Vietz, N. Jazdi, and M. Weyrich, A survey on long short-term memory networks for time series prediction, Procedia CIRP, vol. 99, pp. 650–655, Jan. 2021.

[4]A. Bozanta et al., Time Series Clustering for Grouping Products Based on Price and Sales Patterns. arXiv, Apr. 18, 2022.

[5]S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, Time-series clustering – A decade review, Inf. Syst., vol. 53, pp. 16–38, Oct. 2015.

[6]A. Javed, B. Lee, and D. Rizzo, A benchmark study on time series clustering, Mach. Learn. Appl., vol. 1, Sep. 2020.

[7]F. Mart ínez-Álvarez, A. Troncoso, J. Riquelme, and J. Aguilar-Ruiz, Energy Time Series Forecasting Based on Pattern Sequence Similarity, Knowl. Data Eng. IEEE Trans. On, vol. 23, pp. 1230–1243, Sep. 2011, doi: 10.1109/TKDE.2010.227.

[8] Yuhan Li Beijing University of Technology, Beijing 100083, China , Stock Price Prediction based on Multiple Regression Models,2023.

[9] Mojtaba Nabipour, Pooyan Nayyeri, Hamed Jabani, Amir Mosavi, Deep learning for Stock Market Prediction,2019.

[10] Xiaonan Zuo , 2nd International Conference on Financial Market and Enterprises Management Engineering (FMEME 2023), vol. 44, p. 691-692, Jun. 2022.

[11]M. Chen, N. Narwal, and M. Schultz, Predicting Price Changes in Ethereum.

[12]R. Sarpong-Streetor, R. Sokkalingam, O. Mahmod, H. Daud, and D. Asamoah Owusu, ARIMAX Modelling of Ron97 Price with Crude Oil Price as an Exogenous Variable in Malaysian, 2022, pp. 679–691.

[13]S. Alahmari, Using Nonlinear Machine Learning Algorithms to Predict the Price of Cryptocurrencies, Feb. 2020.

[14]What Is an ARIMAX Model?, 365 Data Science, Jun. 25, 2020. (accessed Jun. 21, 2023).

[15]R. Chow, K-Nearest Neighbors Algorithm: Classification and Regression Star, History of Data Science, Aug. 27, 2021.

[16]M. Chen, N. Narwal, and M. Schultz, Predicting Price Changes in Ethereum.

[17]R. Sarpong-Streetor, R. Sokkalingam, O. Mahmod, H. Daud, and D. Asamoah Owusu, ARIMAX Modelling of Ron97 Price with Crude Oil Price as an Exogenous Variable in Malaysian, 2022, pp. 679–691.

[18]S. Alahmari, Using Nonlinear Machine Learning Algorithms to Predict the Price of Cryptocurrencies, Feb. 2020.

[19]A. C. Harvey, ARIMA Models," in Time Series and Statistics, J. Eatwell, M. Milgate, and P. Newman, Eds., in The New Palgrave. London: Palgrave Macmillan UK, 1990, pp. 22–24.

[20]F. Islam and M. A. Imteaz, The effectiveness of ARIMAX model for prediction of summer rainfall in northwest Western Australia, IOP Conf. Ser. Mater. Sci. Eng., vol. 1067, no. 1, p. 012037, Feb. 2021.

[21]A guide on regression error metrics (MSE, RMSE, MAE, MAPE, sMAPE, MPE) with Python code, Amir Masoud Sefidian - Sefidian Academy, Aug. 18, 2022.

[22]Basic Understanding of ARIMA/SARIMA vs Auto ARIMA/SARIMA using Covid-19 Prediction, Predict the future, Jun. 28, 2020.