ISSN: 1803-3814 (Printed), 2571-3701 (Online)
https://doi.org/10.13164/mendel.2023.2.283

Initial Coin Offering Prediction Comparison Using Ridge Regression, Artificial Neural Network, Random Forest Regression, and Hybrid ANN-Ridge

Toai Kim Tran[1,2,✉], Roman Senkerik[3,✉], Vo Thi Xuan Hanh[1], Vo Minh Huan[1], Adam Ulrich[3], Marek Musil[3], and Ivan Zelinka[2]

[1]Ho Chi Minh University of Technology and Education, Vietnam
[2]VSB-Technical University of Ostrava, Ostrava-Poruba, Czech Repulic
[3]Tomas Bata University, Zlin, Czech Republic

toaitk@hcmute.edu.vn✉, senkerik@utb.cz ✉, hanhvtx@hcmute.edu.vn, huanvm@hcmute.edu.vn, tran.kim.toai.st@vsb.cz, ivan.zelinka@vsb.cz

Abstract

Can machine learning take a prediction to win an investment in ICO (Initial Coin Offering)? In this research work, our objective is to answer this question. Four popular and lower computational demanding approaches including Ridge regression (RR), Artificial neural network (ANN), Random forest regression (RFR), and a hybrid ANN-Ridge regression are compared in terms of accuracy metrics to predict ICO value after six months. We use a dataset collected from 109 ICOs that were obtained from the cryptocurrency websites after data preprocessing. The dataset consists of 12 fields covering the main factors that affect the value of an ICO. One-hot encoding technique is applied to convert the alphanumeric form into a binary format to perform better predictions; thus, the dataset has been expanded to 128 columns and 109 rows. Input data (variables) and ICO value are non-linear dependent. The Artificial neural network algorithm offers a bio-inspired mathematical model to solve the complex non-linear relationship between input variables and ICO value. The linear regression model has problems with overfitting and multicollinearity that make the ICO prediction inaccurate. On the contrary, the Ridge regression algorithm overcomes the correlation problem that independent variables are highly correlated to the output value when dealing with ICO data. Random forest regression does avoid overfitting by growing a large decision tree to minimize the prediction error. Hybrid ANN-Ridge regression leverages the strengths of both algorithms to improve prediction accuracy. By combining ANN's ability to capture complex non-linear relationships with the regularization capabilities of Ridge regression, the hybrid can potentially provide better predictive performance compared to using either algorithm individually. After the training process with the cross-validation technique and the parameter fitting process, we obtained several models but selected three of the best in each algorithm based on metrics of RMSE (Root Mean Square Error), R^2 (R-squared), and MAE (Mean Absolute Error). The validation results show that the presented Ridge regression approach has an accuracy of at most 99% of the actual value. The Artificial neural network predicts the ICO value with an accuracy of up to 98% of the actual value after six months. Additionally, the Random forest regression and the hybrid ANN-Ridge regression improve the predictive accuracy to 98% actual value.

Keywords: *ICO, Prediction, Multi-correlation, Ridge Regression, Linear Regression, Artificial Neural Network, Random Forest Regression, One-hot Encoding.*

Received: 10 November 2023
Accepted: 14 December 2023
Online: 14 December 2023
Published: 20 December 2023

1 Introduction

This paper aims to supply a comprehensive analysis of data analysis techniques and compare baseline techniques for various applications. In today's data-driven world, understanding and making sense of data is essential for decision-making and achieving insights. In addition, this paper focuses on involving specific data analysis techniques in a dataset and evaluating the performance of baseline techniques in different scenarios. The results and findings presented in this paper serve as a starting point for further exploration and development of advanced data analysis techniques, descriptive statistics, and application computational complexity. When a company wants to issue its own cryptocurrency, it usually creates a certain number of tokens, a form of electronic stock, and sells them to investors in the public offering to raise capital. The company will further invest in and develop the finalized product with the proceeds obtained during the public offering. ICO represents a method of fundraising based on the trust of the investors' communities and the finalized products' potential. ICOs were created to solve the problem that more and more startups have good ideas or breakthrough technology but have limited initial capital or limited access to financing channels; these companies will look for ICOs to solve their initial capital and investment demand [19].

However, not all of the ICOs are trustworthy, as there are cases of fraudulent projects or unprofitable projects with investment returns that do not meet investors' expectations. Investors usually wait for approximately 25 and 30 weeks to decide whether to leave these investment channels if the optimized values do not meet their expectations [24]. If investors choose careful ICOs with real potential projects to successfully launch their products, and the tokens are listed on the cryptocurrency market and widely accepted, it can indeed lead to significant development potential. In that case, the token value will increase significantly compared to the cost when investors purchased it at the time of issue. At this time, investors tend to sell these tokens to make the optimal values [19, 7, 3]. Investing in the ICO (Initial Coin Offering) can indeed provide opportunities for small individual investors to potentially gain significant returns with a relatively small amount of capital. ICOs are a crowdfunding method used by blockchain projects to raise funds by issuing and selling their own tokens or cryptocurrencies. These tokens are typically offered at a lower value during the ICO phase, with the expectation that their value will increase once the project is developed and the tokens are listed on exchanges. The future value of the token can increase exponentially in the cryptocurrency market [8, 6]. However, the ICO investment offers little financial guarantee because of its uncertainty about the feasible ability of the business model and trading. As already mentioned, not all of the ICO investments are promising; destructive projects still appear in the community [19]. The best way to overcome the risks of investing in ICO is to analyze and evaluate ICO in every contribution aspect to project success. Several critical factors affect successful ICO investing, which we list below [18, 8]:

- White paper is a public document available on the ICO's website. The white paper description shows the capacity of ICO project success. The ICO fails if the founder team cannot publish a white paper.
- Quality of the issuer team: With many years of experience and good discipline, the team participates in product quality improvement.
- Information about ICO: Information such as token sale-start and end-date, how to trade, price, total supply, and market capitalization show us a guarantee of successful investing.
- Product idea: The company wants to achieve milestones when launching the product (such as what technology, platform, and service are used).
- Famous and influential people in social media often discuss ICO investing topics on social networks such as Facebook, Twitter, and YouTube.
- Opinions of experts in the field of cryptocurrencies.

Additionally, more detailed factors influence uncertainty about ICO success, but social media for an ICO project stands out [3, 18, 19].

Based on the above-mentioned facts, it is clear that ICO value prediction is important, especially for investors and consultants. Value prediction utilizing machine learning is becoming a new global trend, given its accuracy and efficiency in forecasting applications [18]. Many algorithms are used for prediction, such as multiple linear regression, Ridge regression, Artificial neural networks (ANNs), Support vector machine (SVM), etc. [11, 23, 21]. Predicting the upcoming ICO value shortly is also fundamental. When companies, investors, or consultants get a specific prediction with acceptable or high accuracy, they will take appropriate steps based on the prediction result.

The rest of this research paper is organized as follows. Related works, the definition of motivation and originality are followed respectively by Section 2 and 3 presenting the data gathering method and the proposed algorithms for gathering and processing the accessible ICO data. Section 4 describes the data analysis while the data analysis is represented in Section 5. In the next stage of our research, the theoretical basis for building the Ridge regression model based on the linear regression model, Artificial neural network architecture, Random forest regression model, and the hybrid ANN-Ridge regression is represented in Section 6. The empirical findings and evaluations are given in Section 7. Finally, Section 8 concludes and discusses the obtained results.

2 Related Work

In recent years, machine learning has become a new trend and also an effective tool in ICO value prediction tasks. At the same time, prediction accuracy is a challenge when investigating machine learning algorithms [20, 23].

Various machine learning methods have been applied to predict the success of ICO projects, together with an emotion-based analysis to assess its attractiveness [9, 4]. More specifically, these research studies collect user sentiment data based on their comments on Twitter to gauge the success of an ICO and the amount of successful crowdfunding. The methodology was based on Logistic regression and Random forest because of their high accuracy in analyzing user emotions. To predict ICO success, there is another analysis involved based on the natural-language processing model that analyzes terms commonly used in successful ICO white papers for comparisons and evaluations with others. Results show that sentiment analysis was a valuable technique for evaluating the attractiveness

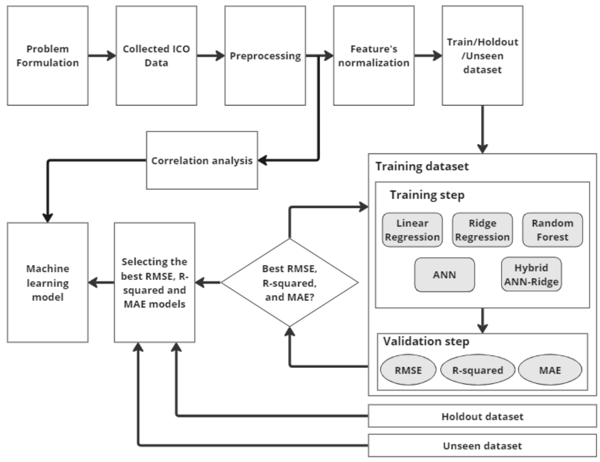

Figure 1: The overview structure of the proposed framework.

of ICO. Thus, Twitter-based comments are supposed to help assess the success of ICO and the amount of fundraising [4, 14]. Moreover, the white paper analysis investigates terms inside papers, with repeating terms stored and classified based on the success of the ICO. The text-mining algorithm is used to classify successful or unsuccessful ICOs [24]. Similarly, another text-mining-based technique has utilized the K-nearest neighbor (KNN) algorithm as a simple machine learning classifier for making precise investment decisions based on examining a project [23]. Research shows that successful ICOs have well-informed white papers with terms that are considered model parameters [24]. Moreover, the performance of the predictive model was validated using three continuous error measurement metrics: Mean Absolute Error (MAE), Maximum Error, and Root Mean Square Error (RMSE) [16]. These metrics are commonly used to assess the accuracy of predictive models. The number of training data increased; the model's performance improved. This improvement is indicated by a tendency for the error in forecasts to decrease. In other words, as more data was used to train the model, it became more accurate in making predictions, resulting in smaller errors between the predicted values and the actual values. By considering these metrics, it can be inferred that the predictive model demonstrated better performance with increased training data, leading to improved accuracy in its forecasts [16]. The use of standard statistical parameters implies that various metrics were employed to assess the models' performance. These metrics might include measures such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R-squared value, or other appropriate evaluation metrics for regression tasks. By comparing these metrics between the models, it was determined that the Random forest method yielded better predictions of the CBR (California Bearing Ratio) value compared to the M5P model. Additionally, based on the statistical parameters and sensitivity analysis, the Random Forest method was found to outperform the M5P model in predicting the CBR value [25].

3 Originality and Motivation

This research aims to forecast the ICO value six months after being released to gauge its success, taking into consideration many factors that influence the value. ICOs can be beneficial or unfavorable as a result. After six months of investment, investors expect the return value to satisfy the rate of return. The motivation was to investigate the effect of individual features from the available ICO data on the overall prediction accuracy and to select and compare simple predictors as a compromise between interpretability, computational complexity, and accuracy. The originality can be defined as follows:

- The paper proposes a methodology for collecting the ICO dataset, data analysis, and further preprocessing of data.
- Based on the data input as the parameters of the forecasting model, this study analyzes the correlation of the inputs compared to the outputs to evaluate the influence of the parameters on the ICO value outcome. The data set consists of 12 fields, covering the main factors that affect the value of an ICO.
- The four methods consisting of Ridge regression, ANN, Random forest regression, and hybrid ANN-Ridge regression are investigated here to evaluate the accuracy of each model's prediction based on the database collected from reputable websites about ICO cryptocurrency.
- The above-listed machine learning models have not been applied to ICO value forecasts about different factors affecting ICO values. Previous studies on predicting the success of ICOs often focused on analyzing user emotions affecting the success of ICOs or relying on white reports to make forecasts. In addition to these factors, the ICO value depends on many other factors that have not been examined to investigate the influence on the accuracy of the ICO value predictions in various simpler machine learning models.

Figure 1 shows the overview structure of the proposed framework. The proposed framework aims to find a machine-learning model to predict the price of ICO after a 6-month release. The development of machine learning models is conducted in several steps. First of all, the problem formulation defines the task of predicting the price of a cryptocurrency token after six months of its ICO. This involves understanding the factors that affect ICO token prices and developing a predictive model based on available data. The objective is to create a model that accurately predicts the ICO price. The model should provide insights into the factors driving token prices and help investors make informed decisions. Next, data collection gathers relevant data related to ICOs including information about the token, ICO duration, USD price, and any other factors that may affect token prices. Then, the data preprocessing step is implemented by preprocessing the collected data by handling missing values, and outliers. Next, preprocessed data is normalized. This step ensures the data is suitable for training the model. The correlation analysis is conducted to identify the variables that have a significant impact on ICO token prices. The result of this correlation analysis is used to determine the key features that affect ICO prices. Next, the collected dataset is divided into training, holdout, and unseen datasets. We compare five machine learning algorithms consisting of linear regression, Ridge regression, Artificial neuron network, Random forest, and hybrid ANN-Ridge regression. We use a training dataset to train these algorithms and evaluate trained models. An iterative process is performed to find the best performance model. After training, the model is considered by the evaluation step to find the best performance model based on RMSE, R-squared, and MAE metrics. The holdout (testing) dataset and unseen dataset evaluate its selected model performance. Finally, a comparison result is analyzed to choose a suitable model to predict ICO prices. By following these steps and conducting a thorough analysis, we can develop a model that predicts ICO token prices and gain insights into factors that impact the success of ICOs.

4 Data Gathering

The developed ICO value forecasting methodology has two main parts: the dataset and the prediction (regression) algorithms. This section describes in detail the process of collecting data from cryptocurrency websites. The sequence of the data processing steps determines the originality of the presented research.

Regarding the data set creation, the data has been carefully selected and filtered (removing invalid data). The collected dataset contains the maximum amount of available and relevant information related to ICOs. As depicted in Figure 2, we retrieve data (in a raw data format) from reputable cryptocurrency sites such as tokendata.io, icodata.io, and coinmartketcap.com through an API that provides valuable information for the analysis and modeling tasks, this API is written by C# as the programming language, and we leveraged the capabilities of C# to interact with the APIs and process the data.

The first phase is the raw data preprocessing before performing the data analysis. In data preprocessing, the steps include removing irrelevant and invalid data, filling in missing data fields/completing missing data fields, transforming data format according to problem requests, and storing data in the database. After collecting the data set, we investigated the invalid data; the term invalid data is data with the *start_date* and *end_date* fields with incorrect values. The lost data is the data where fields like *price_btc*, *market_cap_usd*, *available_supply*, and *total_supply* are missing their values. To handle these two cases of invalid and lost data, we have manually looked up the data on the cryptocurrency websites. During the search, we have encountered many cases of conflicting data. Thus, we decided to gather data from many different sources, find common data between websites, and add them to the data set. The complete dataset has 109 ICOs consisting of 109 rows and 12 columns. The dataset can be exported to a .csv file as input for the prediction algorithms (models).

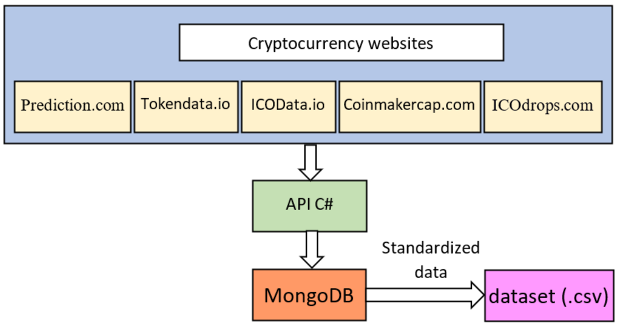

Figure 2: The process of data collection.

Table 1: Dataset structure.

Cross data		Holdout data	Unseen data
Training data	Validation data		
68 ICOs	17 ICOs	21 ICOs	3 ICOs

One-hot encoding is the process of converting categorical variables to a binary form that can be fed to the machine learning model to perform better predictions. However, applying one-hot encoding for classifying the data fields will increase the number of inputs. In this research, we apply the one-hot encoder to encode the alphabet input to the last 4 data fields, which are totally converted into 128 inputs (see the next section) *ICO Duration Day*, *Date ICO launch*, *Month ICO launch*, and *Country ICO* (i.e. the country where ICO was issued). In more specific details, one-hot encoding aims to perform better predictions because these last four fields contain literal data. These data must be converted from alphabet form to numeric form. Therefore, the number of rows is not changed, and the number of columns in the dataset is extended (from 12 to 128).

To get the best possible results, as shown in Table 1, we have separated our dataset as follows:

- Training data - this data was used to train the models
- Validation data - this data was used to validate the prediction
- Holdout (testing) data - This data was applied to evaluate the algorithm's accuracy
- Unseen data - This data was not used during the training or testing phase of the machine learning model that the model has not encountered before. Unseen data is used to assess the model's performance in real-world scenarios and evaluate its generalization capability.

Specifically, the 109 ICOs dataset includes 106 ICOs used to train the model, and 3 ICOs used as a test set (unseen data) to evaluate the algorithm's accuracy. 106 ICOs are divided into two parts: *cross-data* and *test holdout data*, with the rate of 80% for the *cross-data* (including 85 ICOs) and 20% for *test holdout data* (including 21 ICOs). The function of the *cross-data* set is to train the Ridge regression algorithm. The *cross-data* set (including 85 ICOs) is split into two subsets: training data (training set) and validation data (validation set), with the rate of 80% for the training set (including 68 ICOs) and 20% for the validation set (including 17 ICOs) to perform the evaluation. Such validation procedures may avoid overfitting and find better hyperparameters. Testing the accuracy between the predicted and actual values uses model evaluation methods such as RMSE, MAE, and R^2 (see details in section 6).

The *cross-data* set is used to train the models. The result of the training will generate three models with the three best values of RMSE, MAE, and R^2 in the first round. The *test holdout data* set is applied to monitor the accuracy of the complex loop to adjust the parameters and find the best training model based on the model evaluation metrics. Each iteration will use a different sample, so after calculating the model's RMSE, MAE, and R^2 values, the model is changed to gain the best values with the last iteration. The *test holdout data* set is then put into this best model for the calculations of the performance metrics.

5 Data Analysis

Detailed data analysis was performed to examine the individual ICO features, their mutual influence and correlation, and to support decision-making on the choice of the prediction model. The preliminary tests with the collected data set revealed that the machine learning models encounter phenomena such as multicollinearity and overfitting. Thus, in order to confirm the findings, the important step was to analyze the strong and weak correlation between the inputs and the output using the scatter plot and calculate the corresponding correlation coefficient between the inputs and the output.

Figure 3 shows an overview of the correlation between the inputs and the outputs variables and between the inputs variables themselves. These variables are analyzed in correlation as follows.

- *Price USD* is the value of 1 token calculated based on the US dollar. As the correlation coefficient shows, the variables *price_usd* and *price* (output) have a strong relationship (0.88), and the positive correlation means, i.e., assuming an increase in *price_usd*, it will lead to an increase in the ICO value (output).
- *Price BTC* is the value of one token in Bitcoin. The relationship between the variables *price_btc* and *price* (output) is strong (0.88). Similarly, the positive correlation means, i.e., assuming an increase in *price_btc*, this will lead to a growth in the ICO value (output) and vice versa.
- *Total Supply*, also known as *Max Supply*, is the total number of tokens supplied to the market. For example, 0x has 1 billion tokens, and Aeron has 20 million tokens. Based on the correlation coefficient of -0.05, it can be seen that this variable has almost no correlation with each other.
- *Market Cap* is the total market capitalization. The correlation coefficient between two variables *market_cap_usd* and *price* (output) is 0.24. It can be seen that these two variables have a weak relationship.
- *Available Supply* is the number of tokens mined and trading in the market. Based on the graph, the correlation coefficient between the two variables *available_supply* and *price* (output) is -0.05. It can be seen that the two variables have almost no correlation with each other.
- *USD raised* is the USD received by the issuing company, which is summed from the beginning of issuance to the end of the sale. Based on the graph, the correlation coefficient between the two variables *usd_raised* and *price* (output), as shown in Figure 3, is 0.24, which means that the relationship between the two variables is weak.
- *Ethereum value at launch*: although the tokens operate based on other cryptocurrencies such as Ethereum and Bitcoin, the value of Ethereum at the beginning of issuance only has a negligible effect on the output in the downward direction, i.e., if the value of Ethereum is high, the value-output decreases because the correlation coefficient between the two quantities is -0.366 (weak relationship).
- *Bitcoin value at launch*: the value of Bitcoin at the beginning of issuance only has a negligible effect on the output in the downward direction, i.e., if the value of Bitcoin is high, the output value will decrease because of the correlation coefficient between the two quantities is -0.36565. Similarly, as in the previous case, this is a weak relationship.
- *The Month ICO was launched* is the Month of the ICO release. Based on the graph, this quantity has a negligible effect on the output since the correlation coefficient is 0.2879 (weak relationship).
- *Date ICO* was launched represents the day of the Month the ICO was released. Based on the analysis results, this quantity has little effect on the specific *price* (output), with the correlation coefficient from the chart being -0.134 (i.e. weak relationship in the negative direction).
- *Country ICO was launched from* indicates the country where ICO was issued. Based on Figure 3, it can be seen that the correlation coefficient between the two variables is -0.108 (weak relationship in the negative direction).
- *ICO Duration in day*: how many days it takes the ICO release time. For example, 0x was released in 9 days. Based on Figure 3, it can be seen that

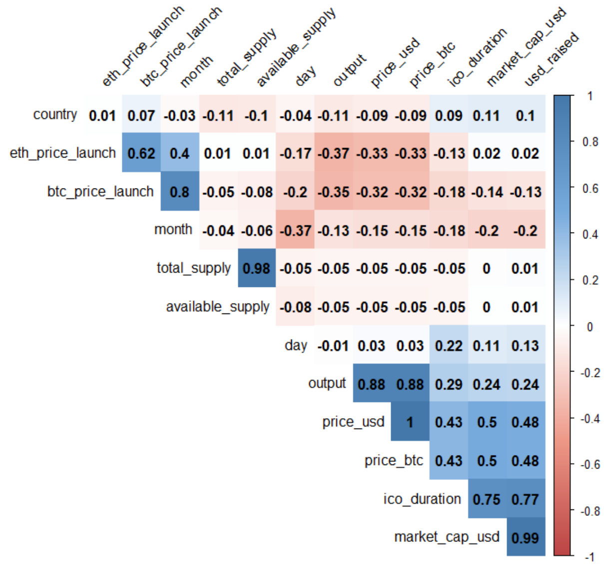

Figure 3: Correlation graph between inputs and between inputs and output.

these two variables can be considered to be almost independent of each other because they have a correlation coefficient of nearly zero.

The given correlation values are subjected to further analysis. As depicted in Figure 3, it is possible to identify which inputs are correlated, such as (*price_usd* & *price_btc*), (*total_supply* & *available_supply*), (*market_cap_usd* & *usd_raised*), (*btc_price_launch* & *month*), (*usd_raised*) & *ico_duration*), (*market_cap_usd* & *ico_duration*).

After analyzing the correlations, we can see the following patterns and relationships. The inputs *price_usd* and *price_btc* have a strong correlation, while the remaining inputs have a weak correlation or there is almost no correlation. If the correlation is high, then a simple linear model can be built for prediction tasks, and it provides good results. However, the weak correlations are significant and there are still correlations between many of the observed values. This drawback means that even though there are two highly correlated inputs, *price_usd*, and *price_btc*, it is impossible to predict the output only by these two inputs and consider the remaining inputs meaningless. Therefore, all inputs must be considered.

The correlation analysis is used to make a selection of variables. A clear demonstration of the differences between values with weak and strong correlations is given in the tables. 2 and 3. Two input variables, *market_cap_usd* and *available_supply*, are chosen because they have a correlation coefficient close to 0 (after rounding). Such a value means that the two variables are independent of each other. Suppose the regression coefficient of *market_cap_usd* is X1 and the regression coefficient of *available_supply* is X2; then SE(X1) and SE(X2) represent the standard error of the regression coefficients X1 and X2. The standard regression error is the coefficient used to measure the accuracy of the estimated regression coefficient. In addition, according

Table 2: The change in the variable values of *market_cap_usd* and *available_supply* after the correlation analysis.

Model	X1	SE(X1)	X2	SE(X2)	Sum Sq
market_cap_usd	6.652e-10	2.576e-10	–	–	73.87
available_supply	–	–	-2.258e-12	4.198e-12	3.39
Both variables	6.653e-10	2.584e-10	-2.265e-12	4.092e-12	73.87/3.41

Table 3: The change in the variable values of *ico_duration* and *usd_raised* after the correlation analysis.

Model	X1	SE(X1)	X2	SE(X2)	Sum Sq
usd_raised	2.580e-09	1.028e-09	–	–	70.06
ico_duration	–	–	2.412e-02	7.755e-03	104.36
Both variables	3.794e-10	1.595e-09	2.188e-02	1.222e-02	70.06/34.92

to the validation results given in Table 2, it follows that with independent inputs, the regression coefficients of the inputs, the standard error of the regression, and the sum of squares (Sum Sq.) are almost unchanged (See the row marked *Both variables*). Another two prognostic variables *usd_raised* and *ico_duration* are chosen because these two variables have correlation coefficients close to 0.77. The analysis and description of the influence of two highly correlated variables are presented below and in Table 3. Similarly to the previous case, the regression coefficient of the *usd_raised* is X1, and the regression coefficient of *ico_duration* X2. SE(X1) and SE(X2) are the standard error of the regression coefficients X1 and (X2). Table 3 clearly shows that the regression coefficients are more significantly changed when using two prognostic variables strongly correlated with each other.

In contrast to two nearly independent variables, the regression coefficients decrease when using highly correlated variables. Specifically, these two highly correlated variables (*usd_raised* and *ico_duration*) are included in the regression model for analysis and evaluation. As a result, they have significantly better differences compared to the regression model for each independent variable. Since the linear regression equation is a function of the model variables and the regression coefficient, if the values of the model variables are large, the regression coefficients will be small to obtain results consistent with the other variables and regression coefficients. Thus, based on the regression coefficient analysis and the standard error of the regression coefficient, the phenomenon of multicollinearity is evaluated for the data in the linear regression model.

Variations in the regression coefficient's standard error can affect the regression coefficient's accuracy. Specifically, if the univariate regression model only uses the *usd_raised* variable, the standard error is 1.0280e-9. However, when the *ico_duration* variable is included in the multiple regression model, the standard error increases to 1.595e-09. Similar to the *ico_duration* variable, the standard error also increases from 7.755e-03 to 1.222e-02. Thus, multicollinearity has occurred in the dataset. In addition, based on the two parameters of training error and test error in Table 4, it can be concluded that there is overfitting, which makes the prediction model misleading.

Table 4: Overfitting analysis in multiple regression model with *test_holdout* set.

	Training Error	Test Error
Model	0.002627	11.545914

Table 4 shows that the training error is 0.002627. Thus, the prediction model results on the training set met our accuracy expectations. These results are achieved by the linear model, which takes the dataset to minimize the difference between the actual value and the predicted value. However, the predicted and actual values do not match well in the test set. The predicted value has varied quite far from the actual value in the test set. By analyzing the training and the test errors in Table 4, we conclude that we have encountered an overfitting phenomenon. This means, that the resulting model is overly influenced by the provided data and is losing its generality. To minimize the impact of this phenomenon, the following technique, as described below, by adding regularization parameters to the loss function is used to address this issue.

6 Methodology

In this section, we will delve into the methodology of each algorithm: Ridge regression model, Artificial neural network, Random forest regression, and a hybrid ANN-Ridge regression. We will discuss and investigate their working principles, parameter-tuning considerations, and practical applications. These algorithms are widely used in various domains, including finance, healthcare, and marketing, to analyze and predict numerical values based on a set of input features.

6.1 Ridge Regression Model

Ridge regression is an extension of linear regression that addresses multicollinearity issues by adding a regularization term to the ordinary least squares objective function. The regularization term, also known as the L2 penalty, controls the complexity of the model by shrinking the coefficients toward zero. Additionally, Ridge regression is a linear regression technique that incorporates a regularization term to address the issue of multicollinearity, which occurs when independent variables are highly correlated. It adds a penalty term to the traditional least squares objective function, which helps in reducing the impact of multicollinearity. The regularization parameter controls the amount of shrinkage applied to the coefficient estimates. Ridge regression provides a balance between model simplicity and accuracy, making it useful when dealing with correlated features. In more detail, Ridge regression is a variant of linear regression developed to overcome the phenomenon of multicollinearity and overfitting [1]. Like linear regression, Ridge regression also tries to fit the data using the residual sum of squares minimization function. The equation of ridge regression has the form (1).

$$I(\theta) = \frac{1}{2N}\left(\sum_{i=1}^{N}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{i=1}^{N}\theta_i^2\right) \quad (1)$$

In equation (1), x^i is the symbol for the inputs of the i^{th} training sample. N is the sample size. y^i is the actual value corresponding to i. The θ values are model coefficients. The parameter λ (called lambda parameter) adjusts the model complexity. This parameter λ is also called the regularization parameter. The λ parameter is being adjusted slightly to avoid overfitting problems while keeping the model's generality at the same time. The parameter λ is often used to evaluate the model's complexity. Larger λ values mean a more complex model. By adding an amount of $\lambda\sum_{i=1}^{N}\theta_i^2$ into the equation (1), which is large when θ is large, the Ridge regression favors models with small θ values. Therefore, non-significant variables with a lower coefficient θ will decrease to zero. The parameter λ is used to correct the model complexity. When λ is large, most model parameters will decrease to zero, leading to the underfitting phenomenon. When λ is small, returning to ordinary linear regression leads to overfitting. Choosing the proper parameter λ is extremely important and may be difficult. We have executed numerous experiments with λ parameter value during training. The best training model also represents the optimum λ parameter value.

6.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are composed of interconnected nodes or neurons, organized in layers: input, hidden, and output layers. Each neuron receives inputs, applies an activation function, and produces an output. In a regression setting, ANNs use a combination of linear and non-linear transformations to learn complex relationships between input features and the target variable. ANNs require careful selection of hyperparameters such as the number of hidden layers, the number of neurons per layer, the choice of activation functions, and the learning rate. These hyperparameters can be tuned using techniques like grid search or random search. ANNs excel at capturing non-linear relationships but can be prone to overfitting if not properly regularized or if the dataset is small. To be more specific, ANNs are a technique suitable for forecasting based on the brain's mathematical model [2]. The ANNs allow the non-linear relationship between input and output variables. Therefore, ANN architecture can be designed as a network of neurons organized in layers. Figure 4 shows the general proposed neural network version forecasting ICO value after six months. In order to determine the number of hidden layers and the number of nodes in the hidden layer, the research conducted repeated experiments with the ANNs model to find the appropriate parameters, giving the most accurate results. The research study used the following parameters for the ICO value prediction model. The input layer, includes 128 data fields, three hidden layers with 100 nodes per each, and the output layer consisting of a single neuron. The Tanh activation function is applied in the hidden layer.

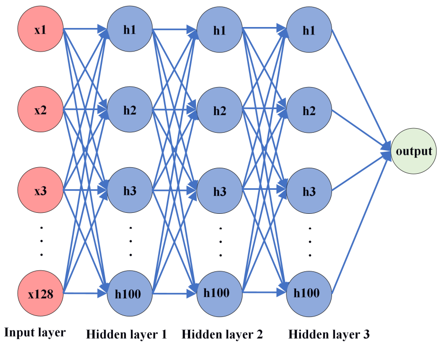

Figure 4: Artificial neural networks architecture for ICO value prediction.

6.3 Random Forest Regression

Random forest regression is an ensemble learning method that combines multiple decision trees to make predictions. This algorithm offers several advantages, including robustness against overfitting, handling of large feature sets, and the ability to estimate feature importance. The number of trees and the depth of each tree are critical hyperparameters that impact the model's performance and computational complexity. In more detail, a Random forest is a supervised machine-learning algorithm that is constructed from decision tree algorithms. The Random forest algorithm consists of many decision trees, and this algorithm performs the outcome based on the predictions of decision trees. It predicts by taking the average or means of output from various trees. Each tree gives a classification and we say the tree "votes" for that class. The forest chooses the classification with the most votes above all trees in the forest. The Random forest is used to solve both regression and classification problems. In addition, Random forest combines many decision trees to provide solutions to complex problems [12] [5]. Thus, it may avoid overfitting. The important thing is that the model needs to optimize turning parameters that characterize the number of features. The Random forest chooses randomly to grow each tree's decisions from bootstrapped data.

6.4 Hybrid ANN-Ridge regression

The hybrid ANN-Ridge regression refers to a combined approach that utilizes both Artificial neuron networks and Ridge regression for prediction. In this hybrid approach, the strengths of both ANN and Ridge regression are leveraged to improve the overall performance and accuracy of the model. Firstly, the ANN is applied to the training dataset. ANN is highly flexible and can learn complex non-linear relationships in data. However, the ANN requires a significant amount of labeled training data and may be prone to overfitting if it is not properly regularized or validated. The regulation techniques consist of L1 and L2 regularization, dropout, and early stopping [13]. Next, the input to the Ridge regression will be the prediction obtained from the ANN. The Ridge regression has been presented to solve the overfitting problem. The Ridge regression reduces the complexity by adding the regulation terms to lower variance in estimating parameters. The Ridge regression is combined with an Artificial neural network that outperforms other methods such as ANN and Ridge regression [22, 17]. The idea is that the ANN will learn to capture any non-linear relationships and patterns in data while incorporating the insights obtained from Ridge regression. By combining the regularization benefits of Ridge regression with the non-linear modeling capabilities of ANN, the hybrid approach can potentially improve prediction accuracy and provide a more robust model.

7 Experiment Results with Multiple Regression Models

The study performs four algorithms consisting of Ridge regression, Artificial neuron network (ANN), Random forest regression, and hybrid ANN-Ridge regression to predict ICO price. The four algorithms are trained and optimized by three evaluation metrics consisting of RMSE (Root Mean Square Error), R^2 (R-squared), and MAE (Mean Absolute Error) [15]. These evaluation metrics are calculated using equations (2), (3), and (4). Let N be the total of historical data given in the dataset. Let y and ŷ be the actual data and corresponding predicted data value. SSR is short for Sum of Squared Residuals which is the sum of the squared differences between the predicted values and the actual values. SST (Total Sum of Squares) is the sum of the squared differences between the actual values and the mean of the actual values.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y-\hat{y})^2} \quad (2)$$

$$R^2 = 1 - \frac{SSR}{SST} \quad (3)$$

$$MAE = \frac{\sum_{i=1}^{N} abs(y-\hat{y})}{N} \quad (4)$$

The training experiment is performed in two steps: the training step and the validation step. The training dataset has 105 ICOs consisting of 68 ICOs for the training step and 17 ICOs for the validation step. One algorithm is trained and optimized by three evaluation metrics consisting of RMSE, R^2, and MAE as shown in Table 5. The training process runs 50 iterations in the Python environment, starting with an untrained model and ending with a trained model, resulting in 50 models. Each model is trained in the criteria of the performance metric. It is easily inferred that every 50 models were produced based on each MAE, RMSE, and R-squared. The Max, Min, and Mean values are computed from these 50 performance metric models. We selected an optimal model from these 50 performance metric models per metric. The study compares four algorithms including Ridge regression, ANN, Random forest, and hybrid ANN-Ridge. As a result, the Ridge regression algorithm achieved three best-optimized models including the best RMSE Ridge regression, the best R^2 Ridge regression, and the best MAE Ridge regression corresponding to minimum RMSE (0.77), maximum R^2 (0.66), and minimum MAE (0.51) in Table 5. Similarly, the ANN algorithm achieved three best-optimized models including the best RMSE neuron network, the best R^2 neuron network, and the best MAE neuron network corresponding to minimum RMSE (0.73), maximum R^2 (0.54), and minimum MAE (0.53) in Table 5. The Random forest performed three best-optimized models including the best RMSE Random forest, the best R^2 Random forest, and the best MAE Random forest corresponding to minimum RMSE (0.14), maximum R^2 (0.93), and minimum MAE (0.1) in Table 5. Finally, the hybrid ANN-Ridge achieved three best-optimized models including the best RMSE hybrid ANN-Ridge, the best R^2 hybrid ANN-Ridge, and the best MAE hybrid ANN-Ridge corresponding to minimum RMSE (0.39), maximum R^2 (0.92), and minimum MAE (0.32) in Table 5. The comparison results are performed by evaluating the 12 optimal models in terms of regression accuracy comparison. Consequently, this will help to accurately analyze and evaluate the proposed regression algorithm.

After the training and selection of these optimal models, the testing step is performed by entering *X_test_holdout* data into each trained model. The training models are loaded to get weights for the testing step. The returned results are the predicted values. The *y_test_holdout* data are actual values. Both the predicted values and actual values are used for performance calculations. The results of the test step are shown in Figure 5 corresponding to four algorithms of Ridge regression, Artificial neuron network (ANN), Random forest regression, and hybrid ANN-Ridge regression, respectively. Each algorithm includes three best models which are trained according to these evaluation metrics. The X-axis shows the ICO name numbering used to compare the actual and predicted values. The Y-axis shows the output price of each ICO. Figure 5 compares the prediction results between *y_test_holdout* which are actual values and *y_pred_holdout* which are predicted values in Ridge regression, ANN, Random forest, and the hybrid ANN-Ridge regression by the best RMSE, respectively. The RMSE metric is chosen because the RMSE is a commonly used metric for evaluating the performance of regression models. While MAE and R-squared are also popular metrics, the RMSE offers several advantages such as sensitivity to outliers, interpretability, and better differentiation [10].

The Ridge regression algorithm in Figure 5 shows the test dataset results based on the RMSE metric in the Ridge regression. A Ridge regressor object is used for Ridge regression which is a linear regression model with L2 regularization. The regressor performs cross-validation internally to determine the best alpha value based on the specified range. The validation data

Table 5: Comparing performance in training dataset using Ridge regression, Random forest, ANN, and hybrid ANN-Ridge regression after 50 repeated runs

	RMSE			R^2			MAE		
Model	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean
Ridge regression	69.81	0.77	3.75	0.66	-580.55	-11.82	20.55	0.51	1.59
Random forest	6.32	0.14	2.11	0.93	-7.00	0.10	2.79	0.10	0.84
ANN	2.98	0.73	1.57	0.54	-6.68	-1.22	8.89	0.53	2.57
hybrid ANN-Ridge	6.56	0.39	2.85	0.92	-15.87	-1.23	3.01	0.32	1.29

Table 6: Key hyperparameters in Ridge regression

Parameter	Description
Alpha=[0.6, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 10]	Constant that multiplies the L2 term, controlling regularization strength.
Solver=svd	The solver parameter determines the algorithm used to solve the optimization problem in Ridge regression
Fit_intercept=True	Whether to fit the intercept for this model.
Normalize=True	The normalize parameter specifies whether to normalize the input variables before fitting the regression model
Random_state=42	The random_state parameter sets the random seed for reproducibility. It ensures that the results are consistent across different runs.

Table 7: Key hyperparameters in ANN regression

Parameter	Description
hidden_layer_sizes=100	This parameter specifies the architecture of the hidden layers in the MLP
max_iter=50	This parameter determines the maximum number of iterations or epochs for which the MLP will be trained.
Activation=tanh	This parameter specifies the activation function to be used in the MLP.
Solver=sgd	This parameter determines the optimization algorithm used to train the MLP
learning_rate=adaptive	This parameter determines the learning rate schedule for the MLP. 'adaptive', which means the learning rate is adjusted during training based on the validation score.
Tol=0.001	This parameter specifies the tolerance for the optimization algorithm. It represents the minimum improvement in the loss function required to continue training.

Table 8: Key hyperparameters in Random Forest

Parameter	Description
n_estimators = 100	The number of trees in the forest
max_depth = None	The maximum depth of the tree.
min_samples_split = 2	The minimum number of samples required to split an internal node.
min_samples_leaf = 1	The minimum number of samples required to be at a leaf node
random_state = 42	Controls both the randomness of the bootstrapping of the samples used when building trees
Bootstrap=True	which means that each tree is trained on a random subset of the training data with replacement
max_features=1	This parameter controls the number of features randomly selected at each split. It determines the subset of features considered for the best split

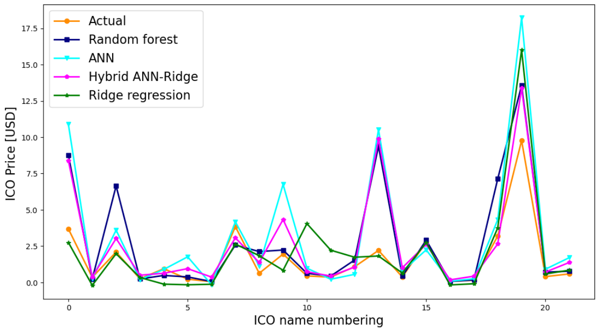

Figure 5: Comparing the test dataset results between *y_test_holdout* and *y_pred_holdout* in Ridge regression, ANN, Random forest, and hybrid ANN-Ridge regression algorithm based on metrics of RMSE.

consists of 17 ICOs. The mode for generalized cross-validation is used to be singular value decomposition. A list of alpha values which are [0.6, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 10], is tested by the Ridge regressor. The alpha is a regularization parameter that controls the amount of shrinkage applied to the regression coefficients. The input features are normalized before fitting the regression model. The input feature is scaled to have zero mean and unit variance. The key hyperparameters are turned to find the optimal combination as shown in Table 6 that uses the Sklearn library.

The ANN algorithm in Figure 5 shows the test

Table 9: Performance comparison using different evaluation metrics on holdout (testing) dataset

Metrics	Ridge regression	ANN	Random forest	Hybrid ANN-Ridge
RMSE	2.5	4.5	2.5	2.17
R^2	-0.14	-1.37	-0.013	-0.95
MAE	1.07	1.33	1.35	1.06

dataset result which is applied to the optimized neural network model for the RMSE metric. The regressor implements a multi-layer perceptron regression algorithm, a neural network-based model. The algorithm is designed to include three hidden layers, 100 neurons in each hidden layer, the Tanh activation function used in hidden layers, stochastic gradient descent solver chosen for weighting optimization. The optimizer iterates over different iterations to find the best model based on the specified metric. Table 7 shows the key hyperparameters configured in ANN.

Similarly, The Random forest algorithm in Figure 5 shows the test dataset results which are applied to the optimized Random forest model. Table 8 describes some key hyperparameters in the Random forest. The Random forest regressor is created with 100 decision trees used in the Random forest. The maximum depth for each decision tree is expanded until all leaves contain less than two samples required to split an internal node. The minimum number of samples at a leaf node is one. The number of features to consider when looking for the best split is one. The parameter controls the number of features randomly selected at each node. The Random forest algorithm generates the same random numbers each time by setting the random state parameter. This helps in obtaining consistent and reproducible results.

The hybrid algorithm's performance is evaluated in Figure 5 using the test data. The basic idea behind such hybrid models is to leverage the strengths of both approaches to improve prediction accuracy and generalization performance. The setting parameters are similar to single ANN or Ridge regression. the ANN is trained including three hidden layers, 100 neurons in each layer, and using a tanh activation function. After obtaining the predictions from the trained ANN models on the training data, Ridge regression is applied to the predictions from the ANN models. The Ridge regression is a linear regression technique that introduces regularization to mitigate overfitting. The mode for generalized cross-validation is also used to be singular value decomposition. The key hyperparameters in this hybrid algorithm are a combination of ANN and Ridge regression shown in Tables 7 and 6.

Table 9 compares the performance evaluation using different evaluation metrics. The comparison result is evaluated using a test dataset with 21 ICOs. The best metric training models are loaded to get weights for testing. The metrics are calculated such as RMSE, R-squared, and MAE to assess the model's predictive accuracy. The RMSE is a measure of the average prediction error between the predicted values and the actual values of these 21 ICOs. It is noticed when comparing the two models, a lower RMSE indicates better performance. The hybrid ANN-Ridge regression algorithm has the lowest RMSE. It suggests that the hybrid model's predictions are closer to the actual values. The lowest RMSE of the hybrid ANN-Ridge algorithm implies the highest accuracy and the best performance of predictions. In terms of the R-squared metric, a higher R-squared value indicates a better fit of the model to the data. The R-squared value of the Random forest algorithm is the highest among the others. The higher the R-squared value, the better the regression model fits the data. In terms of the MAE metric, the MAE measures the average absolute difference between the predicted values and actual values. It represents the average magnitude of errors in the predictions. The lower the MAE, the closer the predictions are to the actual values on average. The MAE of the hybrid ANN-Ridge is 1.06 which is the lowest among the four algorithms. To evaluate the performance of a single algorithm, a lower MAE indicates that the algorithm has more minor average errors and is more accurate in its predictions. It means that, on average, the model's predictions are closer to the true values. In summary, the algorithms of Random forest and hybrid ANN-Ridge regression indicate better performance.

Table 10 shows the predicted results according to three test ICOs selected from 109 ICOs. The test set of three ICOs is an unobserved data set used to evaluate the performance of the three best models in each algorithm using three performance metrics such as RMSE, R^2, and MAE, as mentioned in equation (2), (3), and (4) respectively. If the forecast results are almost as accurate as the actual ICO value after six months of release, the ICO is forecasted with high accuracy. As a result, investors can accurately predict which ICO coins will benefit and satisfy their expectations.

Based on our testing, given our testing data, the four proposed algorithms output pretty accurate results. Especially, the Ridge regression proposed algorithm accurately predicts ICO results with an increase in accuracy from 97% to 99% actual value in Modum ICO.

The proposed ANN model has an optimal R^2 metric for the most accurate prediction results up to 97% of the actual value of Ox ICO. The ANN model gives better prediction results than the Ridge regression but is more computationally heavy and undergoes longer model training time. The Ridge regression predicts an accuracy of 32% to 99%, which is lower than the ANN algorithm model, which achieved an accuracy between 35% to 97%. However, the Ridge regression model is simple and requires fewer hardware resources to train than the ANN algorithm model.

Random forest regression improves predictive accuracy by tuning the hyper-parameters of the algorithm. As shown in Table 10, the Random forest improves the prediction accuracy up to 98% real value by the Crypto20 coin and 96% real value by Modum coin for

Table 10: Comparing the predicted results of three ICOs using four algorithms consisting of Ridge regression, Random forest, ANN, and hybrid ANN-Ridge regression.

	Best RMSE				Best R2				Best MAE			
ICO name	RR	ANN	RFR	Hybrid ANN-Ridge	RR	ANN	RFR	Hybrid ANN-Ridge	RR	ANN	RFR	Hybrid ANN-Ridge
0x(1.08)	1.66 (46%)	1.518 (60%)	1.538 (58%)	1.05 (98%)	1.82 (32%)	1.05 (97%)	1.633 (49%)	0.317 (30%)	1.655(47%)	0.757 (70%)	1.54 (57%)	1.75 (38%)
Modum(2.79)	2.717 (97%)	2.11 (76%)	3.079 (90%)	2.62 (94%)	2.71 (97%)	3.6 (71%)	2.555 (92%)	1.872 (68%)	2.796(99%)	4.596 (35%)	2.912 (96%)	3.28 (83%)
Crypto(0.9786)	0.928 (95%)	0.711 (73%)	0.9 (92%)	0.59 (61%)	1.018 (96%)	0.808 (83%)	0.89 (91%)	1.143 (83%)	1.192 (78%)	0.8 (82%)	0.96 (98%)	1.18 (80%)

the best MAE metric. The accuracy result range is as high as 49% to 98%.

The hybrid ANN-Ridge regression leverages the strengths of both approaches to improve prediction accuracy. The proposed algorithm with the best RMSE metric gives the most accurate ICO prediction results, up to 98% actual value in case of 0x ICO.

The comparison results of these forecasting methods serve as a basis for investors to choose a suitable forecasting method between the Ridge regression, the ANN, Random forest regression, and the hybrid ANN-Ridge regression algorithm. According to the verified experimental results between the four algorithms mentioned above, investors should consider the available hardware resources to determine the appropriate forecasting method when dealing with the ICO value prediction method.

To explain more clearly in correlation results between numbers and percentages in Table 10, assuming that the actual value of 0x coin is 1.08 which is achieved after 6 months of release. Then, this study applies the Ridge regression model, ANN model, Random forest model, and hybrid ANN-Ridge to predict this 0x coin. The prediction results are used to compare and evaluate the models' accuracy. The predicted values are 1.66, 1.518, 1.538, and 1.05 corresponding to the results of applying the Ridge regression model, ANN model, Random forest model, and hybrid ANN-Ridge regression, respectively. The percentage accuracy ratio is used as a performance metric in comparison among models which is described in the following equation (5).

$$Accur\_ratio(\%) = (1 - |\frac{real\_value - pred\_value}{real\_value}|) * 100\% \quad (5)$$

As a result, we can get the accuracy ratio (Accur_ratio) of predicted values which are 46%, 60%, 58%, and 98% in comparison with the real value of 0x coin, corresponding to applying the Ridge regression model, ANN model, Random forest, and hybrid ANN-Ridge algorithm respectively. It is easy to recognize that the hybrid ANN-Ridge model is the best candidate to predict this 0x coin because it gives us the best accuracy ratio among the three models.

8 Conclusion and Discussion

ICO value correlation analysis consists of twelve factors that impact ICO value. The results show that two of the three factors, *price_usd*, and *price_btc*, have a high correlation with output. The remaining variables have a weaker correlation. However, the other input factors are correlated to each other. Thus, the study considers these inputs the primary driver of ICO values. As a basis for a comparative evaluation of the two forecasting methods, the twelve factors that affect the ICO value show that they play an essential role in predictive analysis. The correlation analysis between the factors affecting the ICO value concludes the multicollinearity phenomenon in the linear regression model. This leads to biased results of multiple regression models. The overfitting phenomenon occurs when using multiple regression models. Thus, a technique of adding a component regularization to the error function of the multiple regression model is required to be used, which can reduce overfitting errors. The Ridge regression algorithm is a non-linear regression method that can overcome the challenges of data problems that are not solved by multiple regression. The Ridge regression uses the λ coefficient to adjust the regression coefficient. The 109 ICO collected dataset and the pre-processed data are considered the dataset for the two forecasting models. Of these, 85 ICOs are used for training and evaluation, 21 ICOs are randomly selected to estimate the model's performance ability, and 3 ICOs are randomly selected for the test set. After the training, the best RMSE, best R^2, and best MAE models are selected to predict the test dataset and discover the best predictive model. The forecast simulation results show that the ICO value forecast accuracy after six months is 99% of the actual value using the Ridge regression model with the test set in the case of Modum ICO. Forecasted results show that the ANN algorithm reaches 97% of the actual value with the test set in the case of 0x ICO. The Random forest regression and hybrid ANN-Ridge regression reach 98% the actual value in the case of Crypto ICO and 0x, respectively in terms of forecast accuracy.

Investors expect the expected rate of return when investing in ICO coins. However, the value of ICO coins depends on many factors, primarily when ICO coins are often invested from the time of issuance. Therefore, the nature of risk and the management of risks to successfully invest in ICO is a necessity for investors. In future work, the study will focus on comparing text-mining algorithms to analyze the success of white papers. The white paper acts as a detailed description of the ICO project that a company or a group of developers will implement. The white paper helps investors to better understand and have an overview of the ICO project, thereby deciding whether to invest in this project or not. Based on the prediction, investors could correctly guess whether that ICO white paper is successful or unsuccessful for optimal values.

Acknowledgement: Supported by Internal Grant Agency of Tomas Bata University under project no. IGA/CebiaTech/2023/004, and by the resources of A.I.Lab at the Faculty of Applied Informatics, Tomas Bata University in Zlin, Czech Republic. Further supported by the European Union under the REFRESH — Research Excellence For Region Sustainability and High-tech Industries project number CZ.10.03.01/00/22-003/0000048 via the Operational Programme Just Transition. The following grants are also acknowledged for the financial support provided for this research: grant of SGS No. SP2023/050, VŠB-Technical University of Ostrava, Czech Republic.

References

[1] Adhikari, R., and Agrawal, R. K. An introductory study on time series modeling and forecasting. *arXiv preprint arXiv:1302.6613* (2013).

[2] Babu, C. N., and Reddy, B. E. A moving-average filter based hybrid ARIMA-ANN model for forecasting time series data. *Applied Soft Computing 23* (2014), 27–38.

[3] Burns, L., and Moro, A. What makes an-ICO successful? an investigation of the role of ICO characteristics, team quality and market sentiment. *SSRN Electronic Journal* (2018).

[4] Chursook, A., et al. Can tweets predict ICO success? sentiment analysis for success of ICO whitepaper: evidence from australia and singapore markets. In *15th International Joint Symposium on Artificial Intelligence and Natural Language Processing* (2020), IEEE, pp. 1–5.

[5] Fieri, B., and Suhartono, D. Offensive language detection using soft voting ensemble model. *MENDEL 29*, 1 (Jun. 2023), 1–6.

[6] Fisch, C. Initial coin offerings (ICOs) to finance new ventures. *Journal of Business Venturing 34*, 1 (2019), 1–22.

[7] Hartmann, F., Grottolo, G., Wang, X., and Lunesu, M. I. Alternative fundraising: success factors for blockchain-based vs. conventional crowdfunding. In *2019 IEEE international workshop on blockchain oriented software engineering (IWBOSE)* (2019), IEEE, pp. 38–43.

[8] Hartmann, F., Wang, X., and Lunesu, M. I. A hierarchical structure model of success factors for (blockchain-based) crowdfunding. In *Blockchain and Web 3.0* (2019), pp. 270–308.

[9] Ibrahim, A. Forecasting the early market movement in bitcoin using twitter's sentiment analysis: An ensemble-based prediction model. In *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)* (2021), IEEE, pp. 1–5.

[10] James, G., et al. *An introduction to statistical learning*, vol. 112. Springer, 2013.

[11] Liu, M., Li, G., Li, J., Zhu, X., and Yao, Y. Forecasting the price of bitcoin using deep learning. *Finance research letters 40* (2021), 101755.

[12] Liu, Y., and Wu, H. Prediction of road traffic congestion based on random forest. In *2017 10th International Symposium on Computational Intelligence and Design (ISCID)* (2017), vol. 2, IEEE, pp. 361–364.

[13] Liu, Z., Zhu, Z., Gao, J., and Xu, C. Forecast methods for time series data: A survey. *IEEE Access 9* (2021), 91896–91912.

[14] Lunesu, M. I., and Desogus, O. ICO evaluation websites analysis. In *2020 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)* (2020), IEEE, pp. 48–56.

[15] Mahalakshmi, G., Sridevi, S., and Rajaram, S. A survey on forecasting of time series data. In *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)* (2016), IEEE, pp. 1–8.

[16] Mendoza Uribe, I. Predictive model of the enso phenomenon based on regression trees. *MENDEL 29*, 1 (Jun. 2023), 7–14.

[17] Muayad, A., and Neamah, I. Ridge regression using artificial neural network. *Indian Journal of Science and Technology 9* (08 2016).

[18] Myalo, A., and Glukhov, N. Success of initial coin offering. the empirical evidence from 2016-2019. *The Empirical Evidence from 2019* (2016).

[19] Panin, A., Kemell, K.-K., and Hara, V. Initial coin offering (ICO) as a fundraising strategy: a multiple case study on success factors. In *Software Business: 10th International Conference, ICSOB 2019, Jyväskylä, Finland, November 18–20, 2019, Proceedings 10* (2019), Springer, pp. 237–251.

[20] Peng, Y., Albuquerque, P. H. M., de Sá, J. M. C., Padula, A. J. A., and Montenegro, M. R. The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with support vector regression. *Expert Systems with Applications 97* (2018), 177–192.

[21] Sai, G., and Singh, V. Prediction of compressive strength using support vector regression. *MENDEL 25*, 1 (Jun. 2019), 51–56.

[22] Salim, I., and Hamza, A. B. Ridge regression neural network for pediatric bone age assessment. *CoRR abs/2104.07785* (2021).

[23] Schreiber-Gregory, D. N. Ridge regression and multicollinearity: An in-depth review. *Model Assisted Statistics and Applications 13*, 4 (2018), 359–365.

[24] Serra Deola, B., and Cetingok, B. Analysis of succés factors for initial coin offerings and automatisation of whitepaper analysis using text-mining algorithms. Master's thesis, Universitat Politècnica de Catalunya, 2018.

[25] Suthar, M., and Aggarwal, P. Modeling CBR value using RF and M5P techniques. *MENDEL 25*, 1 (Jun. 2019), 73–78.