Proceedings of the Fourth International Conference on Smart Systems and Inventive Technology (ICSSIT-2022)
IEEE Xplore Part Number: CFP22P17-ART; ISBN: 978-1-6654-0118-0

A Novel Artificial Neural Network Algorithm for Prediction of Natural Gas Prices using Machine Learning

G.Charles Babu[1]
Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous)
Hyderabad, Telagana
charlesbabu26@gmail.com

Kotari Sridevi[2]
Muffakhamja College of Engineering and Technology, Hyderabad, Telangana
sridevi@mjcollege.ac.in

K. Venkata Murali Mohan[3]
TKR Engineering College, Hyderabad, Telangana
kvmuralimohan22@gmail.com

Srisailapu D Vara Prasad[4]
GITAM Deemed to be University, Hyderabad, Telangana
sdvprasad554@gmail.com

J Kavitha[5]
Malla Reddy Engineering College(A), Secunderabad,Telangana
j.kavitha5555@gmail.com

2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT) | 978-1-6654-0118-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICSSIT53264.2022.9716475

Abstract- Natural gas is extensively utilized by the people as the cleanest energy source. With the increase in global population, natural gas has become one of the mainlysignificant energy resources. It has been proposed as one of the solutions to amplify the protection of energy production and reduction environmental pollution about the world. It isn't as it were the most fuel for civilian utilize but too a fundamental input for numerous mechanical applications and power generation. Hence investigate and forecasting natural gas prices have become necessary as it is playing an extended part within the future of worldwide vitality due to its critical natural benefits. Being able to estimate characteristic gas costs benefits different partners and hence it has ended up a really profitable instrument for all showcase members in competitive normal gas markets. Machine learning calculations have ended up well known apparatuses for common gas cost determining. The point of this venture is to examine data-driven prescient models for common gas cost determining based on common machine learning tools such as Artificial neural networks (ANN)

Keywords: Natural Gas Price, Machine Learning, ANN, Multi-class Classification.

1. INTRODUCTION

Natural gas (NG) is solitary of the significant energy resources that aresuitable more admired because of its lower impact on environmental pollution. Predicting the price of natural gas is crucial to manufacturer, providers, dealers and showcase creators who are included within the common gas generation and exchanging. Too the customers are too profoundly affected by the changes within the cost of normal gas. In this manner, request for characteristic gas has expanded significantly in later a long time. According to the U.S. Vitality Data Organization (2016), within the world's essential energy consumption, common gas has the biggest increment, moreover it is anticipated to extend from 120 trillion cubic feet (Tcf) in 2012 to 203 Tcf in 2040. All inclusive, natural gas accounted for 23.8% of essential vitality utilization within the year 2015. Characteristic gas remains the most fuel within the generation of electricity and within the industry division. Characteristic gas cost changes significantly that experiences the partners to a high risk and questionable circumstance. Subsequently more exact estimating makes a difference them to choose a fitting technique in arrange to decrease the dangers. This investigate endeavors to discover perfect way the most perfect way to demonstrate and figure the normal gas costs using Artificial Neural Networks. Important features of neural networks are the capability to be trained and generalization. Neural networks, since of their possessions, offer assistance us to construct a solid and vigorous demonstrate.

2. OBJECTIVES

a. Predicting the price of natural gas could help various stakeholders to take necessary actions in the current marketing situation
b. Helps the market participants in making effective decisions
c. Out of many available machine learning algorithms finding the best suitable algorithm for better predictions
d. To study how the neural networks can be the best model which could give best forecasting results

3. RELATED WORK

Aliyuda Ali from University of Sheffield, UK explained about "Ensemble Learning Model for Prediction of

Natural Gas Spot Price Based on Least Squares Boosting Algorithm”[1].

Anita Takur, Saswat Kumar, AishwaryaTiwari did work on “Hybrid model of gas price prediction using moving average and neural network” which stated “The exactcalculation of gas price is very vital for the manufacturer and customer [2].

Atiq W. Siddiqui from College of Commerce Organization displayed sees on the “Predicting Common Gas Spot Costs Utilizing ARNN show for estimating every day spot gas prices[3]. Bopp (1990) and Hsieh (1990) connected econometrics models, and Pilipovic (2007) executed time arrangement models to anticipate common gas prices[4]. Ogwo, et al. (2007) created an impartial estimating demonstrate to anticipate the normal gas price[5]. Hu and Trafalis (2011) created a unused bit for a neural organize show (vector bolster machine) to foresee the common gas price[6]. Agbon and Araque (2003) connected chaos time arrangement examination and fluffy neural organize show with a nonlinear demonstrate to anticipate the oil and gas costs. Ogwo, et al[7]. Mishra (2012) modeled the characteristic gas cost with time arrangement as well as a nonparametric approach to estimate the price[8]. Reiter and Economides (1999) claim that brief term forecast of gas cost is doable by utilizing slacked factors and neural arrange models [9].

4. METHODOLOGY

The methodology used in this project is Artificial Neural Networks. An Artificial Neural Network learning algorithm, or simply a neural network, or neural net, is a computational learning system that employ a network of functions to understand, analyze the given data input and produce the desired output. The term manufactured neural organize was propelled by human science and the way neurons of the human brain work together to get it inputs from different human faculties. An artificial Neural network (ANN) may be a family of machine learning models that mimic the operation of the human brain and anxious framework. By and large ANNs are depicted as a set of interconnected neurons gathered in a number of layers which exchange signals from one to another. The key concepts in neural organize engineering are layers which are made up of a set of interconnected hubs named neurons containing a yield work named actuation work. The linkage between neurons is called synapses which contain variable numerical weights. Artificial Neural networks essentially comprise of three layers to be specific Input layer, Covered up layer and Yield layer. Input layer, which is the initial layer, accepts the input data (Independent variables) provided by the programmer and output data(dependent variables) are obtained from the output layer which is the last layer. There exist one or more layers, among the input and output layers, known hidden layers. In hidden layers all the calculations are performed to find the hidden patterns and features.

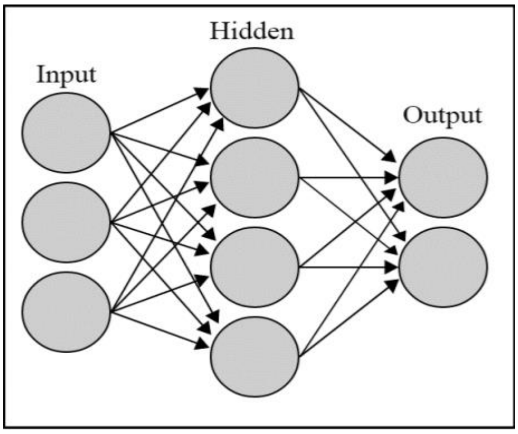

Figure 1 : Schematic of Artificial Neural Network

Each layer comprises of a number of hubs and each hub of a layer is associated to all hubs of the following layer. Each input hub in a layer is increased by a weight, which is able be combined with an actuation work and utilized to determine the layer’s yields. The yield can be a last result or an input to the layer. The weights are initialized by arbitrary numbers and after that prepared by a huge number of accessible information to induce an exact reaction. Distinctive learning calculations are planned to prepare neural systems, which adjust the weights of the neural connections to attain the most excellent fit to the inputs. In this project, we are solving the Multi-class classification problem with Keras.

The model is developed by using the following steps:

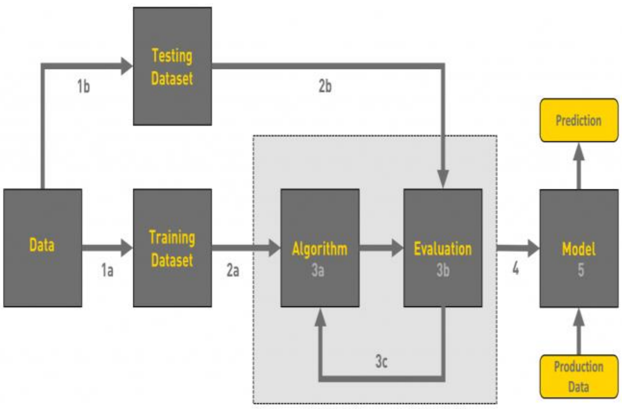

Figure 2 : Flow of building a Machine Learning model

4.1 Data Collection

The dataset is sourced from Henry Hub Natural gas spot price. The dataset contains Daily prices of Natural gas, preliminary from January 1997 to the current year. Prices are in nominal dollars.

4.2 Data Preprocessing

Load the dataset: Import the numpy and pandas libraries and load the dataset.

Taking care of Missing Data: Check if the dataset consists of any missing values. If there are any missing values fill them using the mean value of that column. From this dataset, for classification it is observed that the attribute price falls under 3 categories such as low, moderate, and high prices based on the range of natural gas prices. This results in a Multi-class classification problem with three different class labels.

Data Normalization: Normalize the data by scaling down the values of variables in the range of (0,1) so that the input features have related ranges of magnitude.

Separating the Independent and Dependent variables: Now separate the independent and dependent variables. Here the independent(input) variables are day, month and year and the dependent(output) variable is the price class.

Splitting the dataset into Train set and Testing set: Split the dataset as 80% of the data to train the model and 20% to test the representation. For this we need to import the "train_test_split" from sklearn package. In this splitting we use classes which have attributes like test_size that specifies the percentage of test data and random_state that can have values 0 or 1 which is used to set the test data from the dataset.

Training the ANN model: 4Importing the libraries and packages

We require Keras, data training and model assessment from sci-kit-learn. Keras is a framework used for constructing neural networks using Python. To avoid complexity, it is designed to build a neural network with a few lines of code. It is a high-level system which is depended on tensorflow, theano or cntkbackends. Keras provides an easy front-end layer to build a neural network utilizing TensorFlow. It enables effective and fast experimentation with neural networks. Scikit-learn provides the various tools to pre-process the dataset.

Import keras

Encode the output variable: The output variable comprises of three distinctive string values. When modeling the multi-class classification issues utilizing neural systems, it is nice to reshape the yield variable from a vector that contains values for each lesson esteem to a lattice with a boolean esteem for each lesson esteem or essentially relegate a interesting numerical value to each course esteem. This can be called one hot encoding or making sham factors from categorical factors. In this dataset, the output values are encoded from low, moderate and high to 1, 2 and 3 integer values respectively. Now the data is fully prepared and ready for training. This can be done by to begin with encoding the strings to integrability utilizing the scikit-learn course Label Encoder and after that changing over the vector of integrability to a one-hot encoding utilizing the Keras work to categorical().

Building the Neural Network Model: Firstly, we create our model using the model "Sequential" from Keras. This is used for building the neural network layer-by-layer. Sequential identify to keras that the representation is created successively and the output of every layer we add is input to the next layer we identify. Now define the neural network by adding input, hidden and output layers. To include a layer to the neural arrange model. add is utilized. The Thick is utilized to indicate the completely associated layer. We ought to indicate the contention for what sort of layer we need. The arguments of Thick are yield measurement which is 16 within the to begin with case, input measurement which is 20 for input measurement and the enactment work to be utilized which is rely in this case. The moment layer is comparative, we do not have to be indicating input measurement as we have characterized the show to be sequential so keras will naturally consider the input measurement to be the same as the yield of the final layer i.e 16. Within the third layer(output layer) the yield measurement is 4(number of classes). Presently as we have talked about prior, the yield layer takes diverse enactment capacities and for the multiclass classification problem, it takes softmax activation function.

Adding input layer

The input layer must include the right number of input features. This can be précisedwhile creating the first layer with the input_dim argument and setting it to 3 because there are a total 3 input variables.

$$model.add(Dense(32, input\_dim = 3, activation = 'relu'))$$

Adding hidden layer

Hidden layers are defined with "Rectified Linear Unit" (relu) and 32 neurons each.

$$model.add(Dense(32, activation = 'relu'))$$

Adding output layer

Since this is a multi-class classification issue and there are a total 3 target classes(low-1, moderate-2, and high-3), therefore "softmax" activation function and 3 neurons are used in the output layer.

$$model.add(Dense(3, activation = 'softmax'))$$

Dense – ReLu
Dense layers are fully connected layers in which every neuron is connected to all the neurons in previous layers. Here in each layer 32 nodes are used which means that the fully connected layer has an output size of 32. For a fully connected layer, the ReLu activation function will be used.

Dense – Softmax

For the final fully connected layer the softmax function is to be used. The problem involves a total of 3 classes : low, moderate and high. This is a categorical classification problem where softmax can be used to return a probability over the classes for each input.

Compile Keras Model: Presently the show is characterized, prepared to compile it. Compiling the show employments proficient numerical libraries beneath the backend such as Theano or TensorFlow. Preparing a neural arranges implies finding the leading set of weights to outline inputs to yields within the dataset. We ought to indicate the misfortune work to assess a set of weights and the optimizer is utilized to look through diverse weights for the organize and any discretionary measurements we would like to gather and report amid training. Here cross-entropy is utilized as the misfortune contention. Categorical_crossentropy indicates that we have different classes. This loss is for categorical classification issues and is characterized in Keras as “sparse_categorical_crossentropy“. Characterize the optimizer as the proficient stochastic angle plunge calculation “adam“. This optimizer may be a prevalent adaptation of angle plunge since it tunes itself consequently and gives proficient comes about in a wide run of issues. At long last, since it may be multi-class categorizations issues, gathers and describe the classification accuracy, defined in the metrics argument.

$$model.compile(loss = \text{'sparse\_categorical\_crossentropy'}, optimizer = \text{'adam'}, metrics = [\text{'accuracy'}])$$

Fitting the ANN to the Training set: Presently it’s time to execute the show on preparing information. Ready to prepare or fit the demonstrate on the stacked information by calling the fit() work on the show. Preparing happens over ages and each age is part into a particular number of clumps. One age comprises of one or more bunches, based on the chosen group estimate and the show is fit for numerous ages. The preparing handle runs for a settled number of cycles through the dataset called ages, that must be indicated utilizing the ages contention. Moreover set the number of dataset columns that are considered some time recently the show weights are upgraded inside each age, called the batch size utilizing the batch_size contention. For this issue, we'll run for 300 ages and utilize a group measure of 64.

$$model.fit(x\_train, y\_train, batch\_size = 64, epochs = 300)$$

These setups can be chosen tentatively by trial and blunder based on the yields created. The demonstrate will continuously have a few mistake, but the sum of mistake would be level out after a few point for a given demonstrate setup. This prepare is called model convergence.

Evaluate Keras Model: Evaluate the model using the assess() work and pass the same input and yield of the test dataset. This creates a expectation for each input and yield combine and collects the scores, counting the normal misfortune and any measurements that are already arranged, such as exactness. The assess() work returns a list with two values. The primary esteem is the misfortune of the show on the dataset and the moment esteem is the exactness of the show on the dataset. Hence index [1] is used to get the value of accuracy.

$$model.evaluate(x\_test, y\_test)$$
$$model.metrics\_names[1], scores[1] * 100)$$

Perform Predictions

Make calculations by calling model.predict_classes() function.

Deployment: Firstly set up the environment with the Keras and Tensorflow libraries and then train the model that will expose on the web via Flask. Flask is a micro web framework which is used to deploy Machine Learning models on the web. The trained model is then deployed in a flask. Build a web application which is integrated to the model built. An UI is provided for the users where the user has to enter the values for predictions. The entered values are given to the saved model and prediction is showcased on the UI.

5. ALGORITHMS

There also exists other classification algorithms for solving the Multi-class classification problems along with ANN. Few among them are applied on this dataset and results are examined.

KNN Classifier

KNN organizes the novel data points depending on the correspondence of the previous stored data points. The below lines are applied to the dataset and predictions are made on test data.

```
from sklearn import neighbors
clf = neighbors.KNeighborsClassifier()
clf.fit(train_X,train_y)
# Make class predictions for all observations in X
y_pred = clf.predict(test_X)
```

Random Forest Classifier

Random forest classifier falls under the wide umbrella of ensemble-based learning strategies. They are basic to execute, quick in operation, and have demonstrated to be greatly effective in different spaces. For this import Random Forest Classifier from sk learn ensemble to apply the algorithm on the data.

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_jobs=2)
rf.fit(train_X, train_y)
preds = rf.predict_proba(test_X)[:,1]
y_pred=rf.predict(test_X)
```

Logistic Regression Classifier

Logistic regression is one of the prevalent Machine Learning calculations, which comes beneath the Directed Learning calculations. It is utilized for foreseeing the categorical subordinate variable employing a given set of free factors.

```
from sklearn.linear_model import LogisticRegression
log_reg = LogisticRegression(multi_class='multinomial',solver ='newton-cg')
log_reg.fit(train_X, train_y)
y_pred=log_reg.predict(test_X)
```

Support Vector Machine

The main aim of the SVM methodis to form the leading line or decision boundary that can isolate n-dimensional space into classes so that able to effectively put the unused information point within the adjust category within the future. This best choice boundary is called a hyperplane.

```
from sklearn import svm, datasets
import sklearn.model_selection as model_selection
from sklearn.metrics import accuracy_score
rbf = svm.SVC(kernel='rbf', gamma=0.5, C=0.1).fit(train_X, train_y)
poly = svm.SVC(kernel='poly', degree=3, C=1).fit(train_X, train_y)
poly_pred = poly.predict(test_X)
rbf_pred = rbf.predict(test_X)
```

MLP Classifier

Multi-layer perceptions (MLP) create dominant classifiers that provide high presentationevaluatedby other classifiers, but are regularlycriticized for the number of free parameters.

```
scaler = StandardScaler()
# Fit only to the training data
scaler.fit(train_X)
train_X = scaler.transform(train_X)
test_X = scaler.transform(test_X)

neural_network_class=MLPClassifier(hidden_layer_sizes=(20,10,20))
neural_network_class.fit(train_X,train_y)
predictions = neural_network_class.predict(test_X)
y_pred=predictions
```

6. RESULTS AND DISCUSSION

The accuracy can be termed as closeness of measurements in statistical measures; however it is also used in classifications. In classifications precision is the fraction of true consequencesbetween the total number of cases. The trained model got an accuracy of 86.28% which indicates a good classification accuracy rate.

The output is seen through the user interface where the user enters input values of day, month and year. When the user clicks the predict button it displays the output price class value by considering the input. The below figure is the User Interface made where the user entered a date and the model predicted the Natural Gas Price class value as 1 which means low price on that given date.

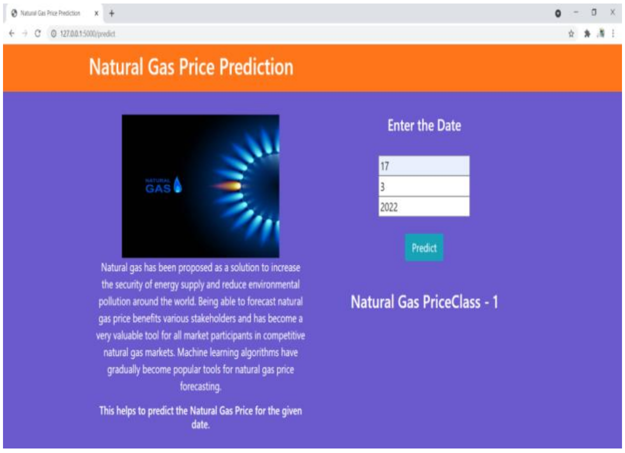

Figure 3 : User Interface that displays Natural gas price class value

Along with the ANN there are some other algorithms utilized to solve the multi-class categorizationissues. They are KNN Classifier, Random Forest Classifier, Logistic Regression Classifier, Support Vector Machine and MLP Classifier. All these algorithms are applied on the dataset and accuracies for each algorithm are obtained.

Classification Algorithm	Accuracy
KNN Classifier	77.525
Random Forest Classifier	82.003
Logistic Regression Classifier	45.959
Support Vector Machine	54.208
MLP Classifier	80.808
ANN	86.287

Table 1: Applied algorithms and the obtained accuracies.

From the above table it can be observed that ANN gave the best accuracy among all the other algorithms. Hence ANN is used to train the model for effective Natural gas price forecasting.

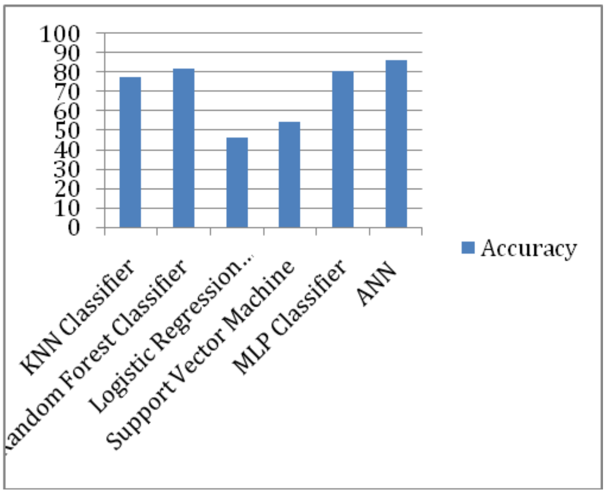

Figure 4 :Graphical representation of displaying performance of Algorithms

The above bar graph shows the comparison of all the applied classification algorithms using the accuracy metrics.

7. CONCLUSION

The present study investigated a method to identify the Natural gas price using a Neural network learning algorithm on the dataset for Multi-class classification problem. Also different classification algorithms are scrutinized in this revision to conclude the best possiblerepresentation. Based on the accuracy, the best model obtained is using ANN. The generated user interface where the user will enter the date for price prediction purpose and displays the desired output. Thus, it is more useful for market participants for effective price forecasting. On the basis of the results that have been produced, the system has provided the best accuracy in the prediction of Natural gas prices. An easy-to-use software package and a user-friendly interface is developed to forecast prices using ANN.

REFERENCES

[1] Salehnia, N.; Falahi, M.A.; Seifi, A.; Adeli, M.H.M. Forecasting natural gas spot prices with nonlinear modeling using gamma test analysis. J. Nat. Gas Sci. Eng. 2013, 14, 238–249. [CrossRef]

[2] Bopp, A. E. (1990). An analytical approach to forecasting natural gas prices. AGA Forecasting Review: American Gas Association, 33-52.

[3] Chandra, V. (2006). Fundamentals of natural gas: an international perspective. PennWell Books.

[4] Haykin, S. (1999). Neural Networks, A Comprehensive Foundation. Prentice Hall.

[5] Hsieh, Y. (1990). Natural Gas Price Projection: A Methodology. American Gas. Association Forecasting Review, 2.

[6] Breiman L. *Classification and Regression Trees*. Abingdon: Routledge; 2017.

[7] 7Ceperi´c, E.; Žikovi´c, S.; ˇ Ceperi´c, V. Short-term forecasting of natural gas prices using machine learning and ˇ feature selection algorithms. Energy 2017, 140, 893–900. [CrossRef]

[8] Su, M.; Zhang, Z.; Zhu, Y.; Zha, D. Data-Driven Natural Gas Spot Price Forecasting with Least Squares Regression Boosting Algorithm. Energies 2019, 12, 1094. [CrossRef]

[9] Lloyd, J.R. GEFCom2012 hierarchical load forecasting: Gradient boosting machines and Gaussian processes. Int. J. Forecast. 2014, 30, 369–374. [CrossRef]

[10] Cossock, D.; Zhang, T. Statistical Analysis of Bayes Optimal Subset Ranking. IEEE Trans. Inf. Theory 2008, 54, 5140–5154. [CrossRef]

[11] Landry, M.; Erlinger, T.P.; Patschke, D.; Varrichio, C. Probabilistic gradient boosting machines for GEFCom2014 wind forecasting. Int. J. Forecast. 2016, 32, 1061–1066. [CrossRef]

[12] Zhang, Y.; Haghani, A. A gradient boosting method to improve travel time prediction. Transp. Res. Part C Emerg. Technol. 2015, 58 Pt B, 308–324. [CrossRef]

[13] Candanedo, L.M.; Feldheim, V.; Deramaix, D. Data driven prediction models of energy use of appliances in a low-energy house. Energy Build. 2017, 140, 81–97. [CrossRef]

[14] Deng, X.; Han, J.; Yin, F. Net Energy, CO2 Emission and Land-Based Cost-Benefit Analyses of Jatropha Biodiesel: A Case Study of the Panzhihua Region of Sichuan Province in China. Energies 2012, 5, 2150–2164. [CrossRef]

[15] Pandian, A. Pasumpon. “REVIEW OF MACHINE LEARNING TECHNIQUES FOR VOLUMINOUS INFORMATION MANAGEMENT.” Journal of Soft Computing Paradigm (JSCP) 1, no. 02 (2019): 103-112.

[16] b. Suma, V., and ShavigeMalleshwara Hills. “Data Mining based Prediction of Demand in Indian Market for Refurbished Electronics.” Journal of Soft Computing Paradigm (JSCP) 2, no. 02 (2020): 101-110

[17] c. Andi, Hari Krishnan. “An Accurate Bitcoin Price Prediction using logistic regression with LSTM Machine Learning model.” Journal of Soft Computing Paradigm 3, no. 3 (2021): 205-217.