International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Forecasting Bangladeshi Garlic Price Using DL and ML Hybrid Approach

MD Tanvir Mahtab [a], Udoy Chandra Dey [b], Md Habibur Rahman Papel [c], Toufiq Hasan Turza [d], Md Imran Mia [e], Md. Tanvir Miah Shagar [f], Lob Kishur Dey [g]

[a,b,c,d,e,f] *Student, B.Sc. in Computer Science and Engineering, Daffodil International University, Bangladesh*

ABSTRACT

Market uncertainty is a constant matter in Bangladesh. Therefore, the prices of our regular substances fluctuate each so frequently. It significantly influences the constituents we consume daily. Almost every meal contains garlic in Bangladesh. People require to keep a record of the price of the materials they use every day but arranging it manually is a strenuous task. Machine Learning (ML) and Deep Learning (DL) permit us to accomplish such a task with much consequence. In this work, we established our dataset from the particulars obtainable from the Ministry of Agriculture, Bangladesh. For predicting the price of garlic, we employed four main architectures based on the DL approach. The first model is the Deep Neural Network (DNN) model, and the second and third models are LSTM-based. Finally, the fourth model is LSTM+ML combined hybrid architecture. By proper analysis, we estimated the best-performing algorithm for the prediction of garlic price. Different matrix results were delivered by the algorithms and performance difference is not so much among all the algorithms. The highest r2 score is achieved by model 3 with a score of 95.83%.

Keywords: Time-series Data Analysis, Garlic Price, Regression, Hybrid algorithm, Prediction

1. Introduction

Many different kinds of spices grow in Bangladesh. Among these spices, garlic is among the most important. Despite a yearly need for 600,000 metric tons of garlic, Bangladesh only manages to produce about 80,000 metric tons of spice [1]. The remaining portion is sourced primarily from India and China, as per government reports. Every passing day sees a rise in the demand for garlic. Thus, the price is going higher since there isn't enough supply to meet demand. Garlic cost between fifty and eighty taka in 2018, as reported by the Bangladeshi Ministry of Agriculture. The year 2019 saw a tremendous surge in the demand for garlic. However, there has been no change to the garlic supply. There was a corresponding rise in price, from 80 to 250 by the year of 2024. Take garlic as an example; on January 1st, 2019, the price was 80 Taka per kg, but by July 14th, it had risen to 180 Taka per kg. There is a lot of focus on this change because of its unusual behavior. The pricing range shows that the increase and decrease are quite sporadic. The impoverished people of Bangladesh cannot bear this expense. The unstructured character of data uncertainty adds complexity to financial forecasting. Predictions are further confounded by the fact that variables such as weather, labor rate, storage clampdown, transportation, and the supply-demand ratio impact the outcomes. Modern AI allows machines to mimic human behavior. Using a variety of ML algorithms, M. M. Hasan et al. [2] successfully eliminated onion market volatility and predicted future onion prices. The possibilities of applying machine learning in finance are vast. To achieve this, we employ the gathered data on garlic prices where we develop some ML and DL models capable of anticipating future garlic prices. Only some of the leading machine learning tools that are available include Scikit-Learn, TensorFlow, Matplotlib, Pandas, and NumPy as observed by Geron et al [3]. In order to use our dataset various feature selection and feature extraction algorithms are used. For the first model, the DNN was used. For the second and third models the type of model that was used was the Long Short-Term Memory (LSTM) model. And at last, the fourth model is a combined architecture of LSTM and ML where the LSTM part is employed only to choose the features, and ML algorithms like Gradient Boosting Regression (GBR), Random Forest Regression (RFR), Linear Regression (LR) are used for training features. Since we are going to be generating predictions of garlic daily price, we employ supervised learning for this. According to the given ML and DL models of the garlic market, it is possible to predict the price of this product at different sources. Our work is focused on this goal.

The following is the outline for this paper. A review of relevant literature is provided in Subdivision II. The suggested approach is detailed in Section III. Unit IV focuses on predicting the price of garlic using several ML approaches and compares the expected results from each. Lastly, our study is concluded with a section that includes a list of potential future works.

2. Literature Review

Forecasting difficulties are a common area of application for machine learning. A lot of effort has been put into using ML to take action about the items' price volatility. This tactic is now much easier to implement thanks to ML.

The implementation phase of an application might guide to increase the exploitation of the basic resources, according to J. Zhang et. al. [5]. To facilitate the supply of resources on demand, this study aims to present a unique template for system-level application resources, demand phase analysis, and vaticination. Utilizing clustering and supervised learning algorithms, they have presented application resource mandate phase analysis and prototypes for prophecy. Their data was collected by keeping tabs on agents that were installed in the virtual machine and stored in an operational database.

To aid efficiency in identifying and anticipating energy demand for each client, D. J. N. Srinivasan [6] created a medium-term energy petition prediction system. They have used ARMA, GMDH, an artificial neural network, a double exponential smoothing layer, a single exponential ironing layer, and a double heartrending average. Compared to older time sequence and regression base models, those based on Artificial Neural Networks (ANNs) be the most effective in terms of producing predictions that are both more precise and require less lever effort. One such model is the Group Method of Data Handling (GMDH), which is based on ANNs. The GMDH neural network predicted a total load series of 1.73 and a combined forecast of 1.52 from the individual series.

According to Z. H. Khan et al. [7], while several methods have been tried to predict stock market prices, none of them have been proven to be an adequate tool for making predictions. Since Artificial Neural Networks (ANNs) are so well-liked for discovering previously unseen specimens in data, they have been employed. This example makes use of a multilayer feedforward network and the backpropagation technique to train the network. The estimated cost and mistake for two input datasets was 401.9 and 1.74%, while for five input datasets, it was 392.7 and 0.58%.

A system was developed by S. F. M. Hussein et. al. [8] using Artificial Neural Network (ANN) algorithms with current gold data temporal successions. Customers will receive forecasts every day from the system. They suggested using a network of radial foundation functions (RBFs) to estimate the price of gold using an objective forecasting method. A Single Radial Basis Function Neural Network (SBFNN) had an SSE value of 269.04, a Multiple Radial Basis Function Neural Network (MRFN) of 155.23, and an Auto-Regressive Model (AR) of 114.05 were the results.

An algorithm for predicting future oil prices was developed by R. Jammazi et al. [9]. An influential prediction of the quantity of crude oil was obtained by running a hybrid model called HTW-BPNN and HTW-MBPNN. They tested the model's adaptability with three different galvanization functions: hyperbolic tangent, bipolar sigmoid, and specifically sigmoid. Results from HTW-MBPNN outperform those from traditional BPNN, on the whole.

To forecast future home prices, B. Park et al. [10] used Machine Learning. The purpose of this research is to enhance the accuracy of property price forecasts by analyzing information from the Multiple Listing Service (MLS) of the Metropolitan Regional Information Systems (MRIS) for 5359 townhouses in Fairfax County, Virginia. A home price projection model employing machine learning algorithms including C4.5, RIPPER, Naïve Bayesian, and AdaBoost has been started, and they are comparing the performance of these algorithms in terms of classification accuracy.

To forecast the clearance prices for the energy markets one day in advance, D. Singhal et al. [11] suggested a neural network approach. The neural network follows the architecture of a three-layer back propagation network. Electricity prices in nonintervention markets are very sensitive to trends in load mandate and reimbursement prices, according to the neural network model's price prediction values.

To predict the indication of the regular price change with the greatest possible accuracy, S. Velankar et al. [12] suggested a technique based on Machine Learning. They have used a Random Forest model in conjunction with the Generalized Linear Model (GLM) and Bayesian Linear Regression. Following the final prediction, the accuracy can be linked to a plethora of models.

Md. M. Hasan et. al. [13] proposed an inconsistency determinator of green chili market price in Bangladesh. They collected the daily price of green chili and divided the price into three classes High, Mid, and low. Finally, they used five machine-learning classification algorithms after preprocessing. In the result comparison stage, they got the highest accuracy 95.34% by Random Forest algorithm using 30% test data. The main limitation of their work is a limited number of data and they acquire daily data only Dhaka region.

There are certain similarities and differences between our work and the previously mentioned works. We aimed to address the volatility of the garlic price market through our work by making predictions about its future values.

3. Methodology

As shown in Figure 1, our research is concluded by an absolute five-step approach. These are the measures to take.

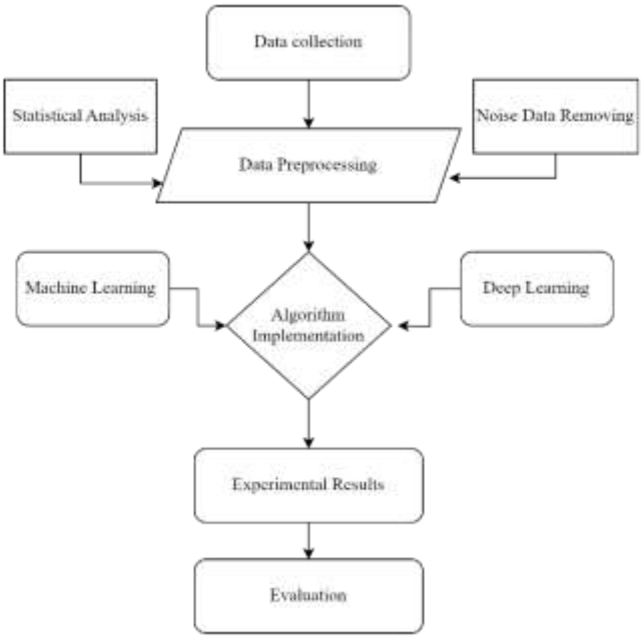

Fig. 1 - Methodology Diagram

3.1 Data Collection

Collecting data is a challenging but necessary part of any research project. The Bangladeshi Ministry of Agriculture's website was the source from which we compiled our garlic pricing. There are two sections to our dataset. Initially, it was used for testing and training purposes. Yet another component was used for forecasting. We utilized 730 daily prices from 2022 and 2023 for training and testing. To make real-life predictions, 60 daily prices from 2024 were utilized.

3.2 Data Preprocessing

It is critical to analyze the collected data in order to determine the most appropriate method for this dataset. We applied two steps in this stage.

3.2.1 Nose Data Removing: The total of this stage is emphasized by null handling. We noticed that some null value is available in the price column we filled it by the mean price approach. But if any field contains an unwanted date or discontinuous date then this field is detected as noise data and we removed it.

3.2.2 Statistical Analysis

Besides, the following Figure 2 displays the frequency distribution table of the garlic prices, and the price varied from roughly 50 to 225 units. Alternatively, on the X-axis, there are prices of the given products, while on the Y-axis, there are frequencies of these prices. From the histograms, it is clear that garlic prices are most common at 75, 100, and 150 among the given varieties as the height of the bars shows. The density curve based on the list of products makes it easier to visualize these clusters and specifically the variation in garlic prices in the market.

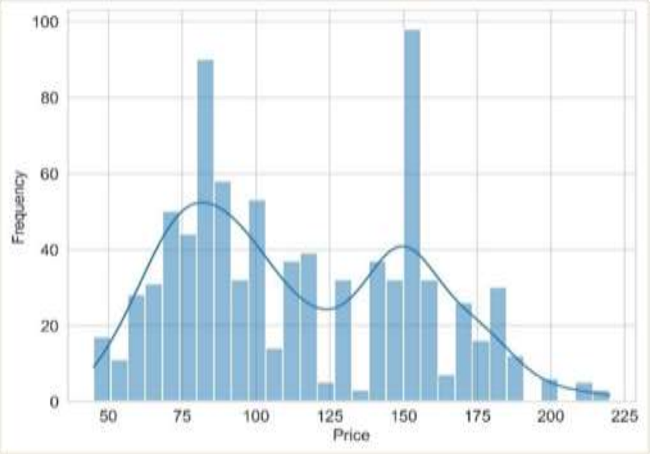

Fig. 2 - Distribution of price

Figure 3 shows the increase of the total prices cognitive over one year in the form of the monthly distribution. The evident trend clearly shows that total prices were relatively high at the initial months, especially around March and April. Thus, the total prices appear to be significantly raised from January (1) through April (4), which may point to the high-demand or 'peak' season of consumption in the first quarter of the year. Afterward, there is a sharp decrease in the total price as from May (5) but a constant average, though lower rates to as August (8). This may also point to a post-peak effect suggesting that there is normalcy or less activity in the market.[14] From September (9) to December (12) which is near year-end, overall total prices have minimal shifts but are, however, lower than the initial parts of the year.

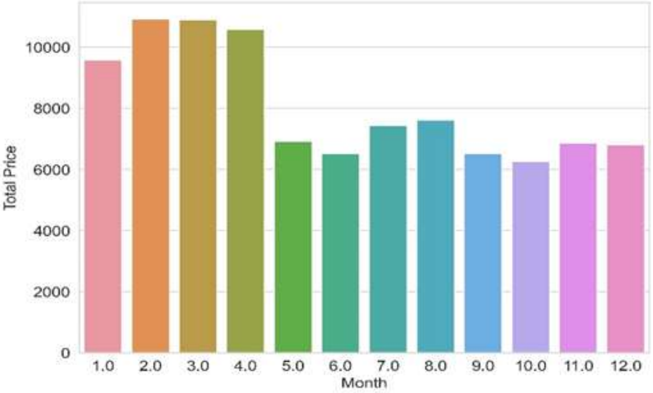

Fig. 3 - Total Price by Month

3.2.3 Algorithm Implementation

This section is divided into two stages, In the first stage we used a total of 3 deep learning algorithms, and in the second stage one Deep Learning and two Machine Learning combined algorithms are used. All algorithm architecture is given below:

Table 1 illustrates the structure of the neural network model 1 To build the model, it begins with an input layer, which has 6 units but does not contain parameters. This is then succeeded by a layer with 256 units and 1792 parameters Dense. Following this is a dropout layer with a dropout of 256 which is an activation function that does not have any parameters. This is followed by the second fully connected dense layer with 128 neurons and 32,896 parameters and the dropout layer with 128 neurons but with no parameters. The model goes further with density containing 64 units and 8,256 parameters. Last but not least, there is a layer that contains only one neuron with 65 parameters. The overall number of parameters in the model is 43,009 and all of them are adjustable parameters, that is, the parameters which can be updated during the training process; there are no fixed or frozen parameters.

Table 1 - Model 1 Architecture

Layer (type)	Output Shape	Parameters
input	[None, 6]	0
Dense	[None, 256]	1,792
Dropout	[None, 256]	0
Dense	[None, 128]	32,896
Dropout	[None, 128]	0
Dense	[None, 64]	8,256
Dense	[None, 1]	65
Total params		43,009
Trainable params		43,009
Non-trainable params		0

Table 2 explains the architecture of a neural network that involves LSTM and the density layer of model 2. The first layer adopted in the construction of the network is an LSTM layer, comprised of 50 units; it has 11,400 parameters resulting from the number of synaptic weights at initialization and the presence of hidden layers. There is a second LSTM layer just after the previous one with 50 units and an output shape of [None, 50] for all layers, which has 20,200 parameters. This is again a dropout layer with the same output shape and zero parameters but the net is not fully connected anymore. The last layer is a dense layer having one unit, the output dimension is None, 1, and the number of parameters is 51. Altogether, the network, observed at the end of the 111th epoch, has 31,651 parameters, each of which is trainable, with non-trainable parameters being nil.

Table 2-Model 2 Architecture

Layer (type)	**Output Shape**	**Parameters**
LSTM	[None, 10, 50]	11400
Dropout	[None, 10, 50]	0
LSTM	[None, 50]	20200
Dropout	[None, 50]	0
Dense	[None, 1]	51
Total params		31,651
Trainable params		31,651
Non-trainable params		0

Table 3 shows how the neural network architecture is constructed of model 3. The network includes as the first layer an LSTM layer, which has 42,800 parameters. Next is the activation layer with 400 followed by a batch normalization layer. The next layer is another LSTM layer which contains 80; to 400 units and from the batch normalization layer which has 400 units, and the last type of layer which is the dropout layer doesn't contain any units. After that it goes through another LSTM layer with 80,400 parameters, then there is a batch normalization layer for normalizing the data with 400 parameters [15], and finally the last dropout layer. The resulting layer is a fully connected layer with 101 units. So, in total, there are 204,901 parameters in the network, out of which 204,301 are trainable parameters and 600 are non-trainable.

Table 3-Model 3 Architecture

Layer (type)	**Output Shape**	**Parameters**
LSTM	[None, 10, 100]	42800
Batch normalization	[None, 10, 100]	400
Dropout	[None, 10, 100]	0
LSTM	[None, 10, 100]	80400
Batch normalization	[None, 10, 100]	400
Dropout	[None, 10, 100]	0
LSTM	[None, 100]	80400
Batch normalization	[None, 100]	400
Dropout	[None, 100]	0
Dense	[None, 1]	101
Total params		204,901
Trainable params		204,301
Non-trainable params		600

Table 4 signifies a neural network model that has LSTM and dropout layers. The architecture starts with an LSTM layer with a parameter number of 69,120 units. After that comes a dropout layer in which some of the nodes within a layer are randomly turned off during training of the neural network. The next layer in the proposed architecture is another LSTM layer which allocates 49,408 parameters, followed by another dropout layer. The last hidden layer is the LSTM layer which utilizes 12,416 parameters. In total, the model has 130,944 parameters: all these parameters are trainable, whereas there are no nontrainable parameters in the model.

Table 4 - Model 4 Architecture

Layer (type)	Output Shape	Parameters
LSTM	[None, 1, 128]	69120
Dropout	[None, 1, 128]	0
LSTM	[None, 1, 64]	49408
Dropout	[None, 1, 64]	0
LSTM	[None, 32]	12416
Total params		130,944
Trainable params		130,944
Non-trainable params		0

In Table 5 we can see that there is no Neural Net for Learning. Because is our hybrid model where the LSTM-based[16] first part is used for feature extraction only, and we trained extracted features by two different Machine Learning Models. ML algorithm implementation is given below:

The table shows the set of hyperparameters of three given machine learning algorithms. All the parameter is chosen by the GridSearchCV algorithm. Finally, for the Gradient Boosting, the setting is the learning rate at 0.1, the maximum depth has to be set to 3, and the number of estimators has to be 100. Also, Random Forest including the maximum depth is set as 5, the maximum number of the features considered in each split is log2, and the number of estimators will be a hundred. We can establish that Linear Regression does not have any relevant hyperparameters used. This overview gives a reader a brief of the configurations of each algorithm, presumably tailored towards enhancing the Algorithm's performance measures.

Table 5 – Hyperarameter Tuning

Algorithms	Hyperparameters
Gradient Boosting	Learning rate = 0.1, max depth = 3, n_estimators = 100
Random Forest	Max depth = 5, max features=log2, n_estimators = 100
Linear Regression	N/A

4. Experimental Result

Table 6 presents the results of several predictive models based on MSE, RMSE, and R² Score indicators. Out of all the models, Model 3 has the lowest MSE of 69. 13 and the RMSE of 8. 31, which means that compared to other models, it has the smallest average squared errors and deviations of the predictions.[17] Moreover, its high R² Score of 0. 9583 hints that it only captures about 95 percent. As evident by the results, although Model 1 and Model 2 show decent accuracy, they are not nearly as accurate or as good as Model 3 when it comes to explaining the underlying mechanisms of the results. Comparing the performances of the LSTM-based models, namely LSTM+GBR, LSTM+RFR, and LSTM+LR for all the metrics, that is, MSE, RMSE, and R² Scores, it can be inferred that all the models have lesser effectiveness in the ability to minimize both the total and mean squared errors, variance explanation as compared with the random forest models. That is LSTM+LR gives the highest mean errors and the lowest R-squared values in other words the worst performance out of all four models. Hence, considering the analyzed metrics, Model 3 is the most suitable choice for PREDICTIVE accuracy and reliability as attainable to mirror variability in the given dataset.

Table 6 – Accuracy Table

Model	Epochs	Best Epochs	MSE	RMSE	R² Score
Model 1	1000	412	72.41	8.59	0.9505
Model 2	1000	460	89.15	9.23	0.94
Model 3	1000	536	69.13	8.31	0.9583
LSTM+GBR	1000	1000	98.98	9.94	0.933

LSTM+RFR	1000	1000	97.45	9.87	0.934
LSTM+LR	1000	1000	100.62	10.03	0.932

The Figure 4 illustrates the loss of a deep-learning model 3 over epochs. Looking at the training loss which is shown by the blue line and the validation loss as shown by the orange line, it is apparent that the model was learning very fast in the initial stages and that the total loss was reducing faster. It can also be observed that during the training the losses settle down and reach near to zero and do not fluctuate much in the subsequent epochs implying that, the model has succeeded in learning about the data set characteristics and is ready to generalize the same for other unseen data as well. This can be observed by comparing the training and the validation set and observing that no overfitting has occurred. This suggests that additional training for a period after 200 epochs will not produce more improvements suggesting a good efficient training period.

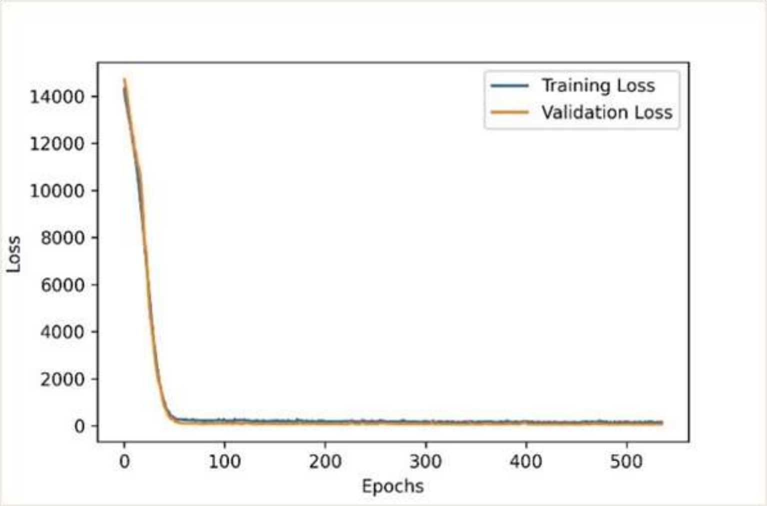

Fig 4 - Training loss vs validation loss of best model 3

Figure 5 represents the actual and predicted values of a set of samples and their comparison. Here, the X-axis is the sample index, which is a range of days, while the Y-axis shows the actual and predicted values. The blue line represents the actual values for each sample while the red line represents the predicted values. As in the case of accuracy, the graph shows that the predicted values are in rather strong agreement with the actual values in certain regions like day 1-9, which means that the model performs well in these regions, while in other regions, certain differences are observable.

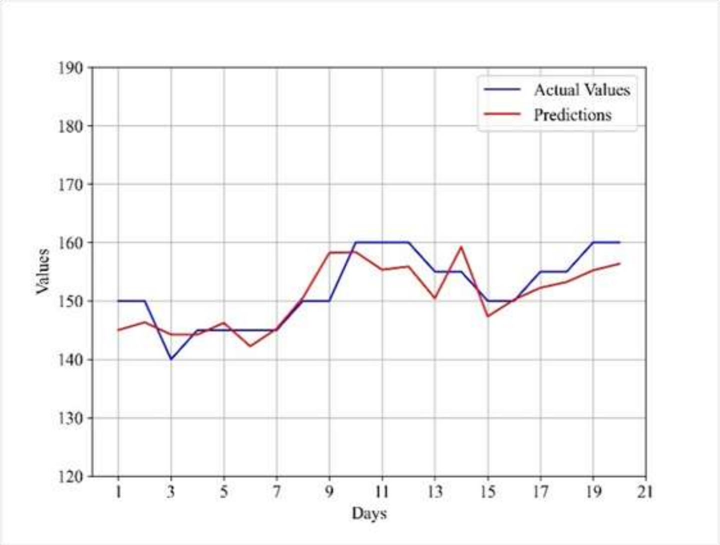

Fig. 5 - Comparison between real and forecasting price

Based on the forecasted real-world data from the first 20 days of January 2024, the evaluation performance metrics are provided in Table VII. In this process, Mean Squared Error (MSE), Root Mean Squared Error (RMSE) & R2 Score are used to measure the performance of the model. For our application, the Mean Squared Error (MSE) is 13. 412; the lower the value the better the performance.[18] From the obtained results, it is possible to mention the RMSE value of 3.66, which points to an error measure that is in the same units as the target variable. This makes the error measure easier to interpret. The r2 score of this model is about 62% which indicates that in the field of real-world applications, these indicators, combined, offer a comprehensive outlook about prediction accuracy and assurance of the model.

Table 7 – Evaluation Result.

Model	MSE	RMSE	R² Score
Model 3	13.412	3.66	0.627

Conclusion

We used three classic ML techniques and four deep-learning models in our work. When we used these methods in our dataset, we found that they did not produce identical results. Our LSTM model 3 architecture achieved the highest efficiency, with a r2 value of 95.83%. So, we'll utilize it to predict how much garlic will cost in the future by this model. Our reliance on data from just two years is the project's biggest shortcoming. We plan to gather further information in the future and create a market intelligence system for Bangladesh that can assess the cost of all essential goods.

References

1. Dailyindustry, “Demand 5-time higher than production in BD,” Jan. 30, 2023. https://dailyindustry.news/demand-5-time-higher-than-production-in-bd

2. M. M. Hasan, M. T. Zahara, M. M. Sykot, R. Hafiz, and M. Saifuzzaman, "Solving Onion Market Instability by Forecasting Onion Price Using Machine Learning Approach," in 2020 International Conference on Computational Performance Evaluation (ComPE), 2020, pp. 777-780: IEEE.

3. A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, 2019.

4. R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, “A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction,” Journal of Applied Science and Technology Trends, vol. 1, no. 2, pp. 56–70, May 2020, doi: 10.38094/jastt1224.

5. J. Zhang, M. Yousif, R. Carpenter, and R. J. Figueiredo, "Application resource demand phase analysis and prediction in support of dynamic resource provisioning," in Fourth International Conference on Autonomic Computing (ICAC'07), 2007, pp. 12-12: IEEE.

6. D. J. N. Srinivasan, "Energy demand prediction using GMDH networks," vol. 72, no. 1-3, pp. 625-629, 2008.

7. Z. H. Khan, T. S. Alin, and M. A. J. I. J. o. C. A. Hussain, "Price prediction of share market using artificial neural network (ANN)," vol. 22, no. 2, pp. 42-47, 2011.

8. S. F. M. Hussein, M. B. N. Shah, M. R. Abd Jalal, and S. S. Abdullah, "Gold price prediction using radial basis function neural network," in 2011 Fourth International Conference on Modeling, Simulation and Applied Optimization, 2011, pp. 1-11: IEEE.

9. R. Jammazi and C. J. E. E. Aloui, "Crude oil price forecasting: Experimental evidence from wavelet decomposition and neural network modeling," vol. 34, no. 3, pp. 828-841, 2012.

A. Park and J. K. J. E. S. w. A. Bae, "Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data," vol. 42, no. 6, pp. 2928-2934, 2015

B. Singhal, K. J. I. J. o. E. P. Swarup, and E. Systems, "Electricity price forecasting using artificial neural networks," vol. 33, no. 3, pp. 550-555, 2011.

10. S. Velankar, S. Valecha, and S. Maji, "Bitcoin price prediction using machine learning," in 2018 20th International Conference on Advanced Communication Technology (ICACT), 2018, pp. 144-147: IEEE.

11. Md. M. Hasan, Md. R. Alam, M. A. Shafin, and M. A. Mithu, “Determining the Inconsistency of Green Chili Price in Bangladesh Using Machine Learning Approach,” in Algorithms for intelligent systems, 2021, pp. 397–406. doi: 10.1007/978-981-16-0586-4_32.

12. A.V. Gevorkyan, “Exchange market pressure and primary commodity – exporting emerging markets,” Applied Economics, vol. 51, no. 22, pp. 2390–2412, Nov. 2018, doi: 10.1080/00036846.2018.1545077.

13. S. Santurkar, T. Dimitris, I. Andrew , M. Aleksander. "How does batch normalization help optimization?." Advances in neural information processing systems 31 (2018)

14. H. Anantharaman, M. Abdullah, and B. T. Shobana. "Modelling an adaptive e-learning system using LSTM and random forest classification." In 2018 IEEE Conference on e-Learning, e-Management and e-Services (IC3e), pp. 29-34. IEEE, 2018.

15. C.Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” PeerJ. Computer Science, vol. 7, p. e623, Jul. 2021, doi: 10.7717/peerj-cs.623.

16. Hasan, M. M., Zahara, M. T., Sykot, M. M., Nur, A. U., Saifuzzaman, M., & Hafiz, R. (2020, July). Ascertaining the fluctuation of rice price in Bangladesh using machine learning approach. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE