[1] Vallabh G. Patel

Vinodray Thumar[2],

Manmitsinh C. Zala[3],

Bhavinkumar Patel[4]

OM P. Mehta[5]

Dhaval J. Varia[6]

Developing A Novel Approach to Forecast the Market Valuation of Newly Manufactured Cars by Leveraging A Comprehensive Array of Vehicular Attributes

Journal of Electrical Systems

Abstract: - Company declares price for new manufactured car based on feature of car and brand name of company. This neural network model can helpful to manufacturer to estimate price of car based on features of car and its own brand name. This model is also useful for buyer to verify price of various new launched car in market. Numerous techniques, including time series analysis, technical analysis, fundamental analysis, statistical analysis, and others, are employed in an effort to forecast the value of newly introduced cars on the market, but none of these techniques has been shown to be a reliable source of information. A popular method of identifying an accurate automotive value based on historical data is to use Artificial Neural Networks (ANNs), a branch of Artificial Intelligence (AI). ANNs consist of two modules: a training session and price prediction based on previously taught data. For the training session, we employed the back propagation technique, and the network model for price prediction was the multilayer feed forward network.

Keywords: Artificial Neural Network (ANN), Artificial Intelligence (AI), Backpropagation, Neural Network (NN), Multilayer Feed-forward Network, Car Price Prediction

I. INTRODUCTION

Every month many of companies launching new car model with variety of features. Due to overall growth of economy car market is growing faster. People have confusion over selection of car from various models which have different feature and price. Buyer can predict price of various model from various manufacturer and calculate historical value of new car using this trained ANN model. So user select best model of car out of multiple available model of car and get value for money spend to by a car. Manufacturer also get helps to decide launching price for new model car. Artificial Neural Networks are comparable to nonparametric, nonlinear regression models, according to evaluations of several statistical models. Therefore, it is possible for Artificial Neural Networks (ANN) to identify intricate and unfamiliar patterns in data, which can be very useful for predicting the price of automobile models. Between 1970 and 2024, we used a collection of more than 90,000 second hand automobiles. This dataset provides a thorough look at the automotive sector, offering insightful information to academics, fans, and business experts alike. These pre-processed data records of different cars with varying attributes and prices. We have trained a multilayer artificial neural network model utilizing this pre-processed data.

Predicting automobile prices has been a topic of great interest in study because it demands a field expert's knowledge and discernible effort. A large variety of unique attributes are looked at in order to make a prediction that is accurate and trustworthy. With machine learning, the right and fair price for a particular used car may be predicted using past data from different vendors and purchasers. To achieve this, train the model with the used car dataset, which contains many variables and parameters including the production year, the number of cylinders, the mileage, whether the car runs on gasoline or diesel, and if the transmission is automatic, manual, or another type.

[1, 3, 5, 6]Assistant Professor, Department of Information Technology, Vishwakarma Government Engineering College, GTU

[2,4]Assistant Professor, Department of Computer Engineering, Vishwakarma Government Engineering College, GTU

Following Features of car are included in data analysis:

- Model: The model of the car.
- Year: The manufacturing year of the car.
- Price: The price of the car.
- Transmission: The type of transmission used in the car.
- Mileage: The mileage of the car.
- Fuel Type: The type of fuel used by the car.
- Tax: The tax rate applicable to the car.
- MPG: The miles per gallon efficiency of the car.
- Engine Size: The size of the car's engine.
- Manufacturer: The manufacturer of the car.

We used this dataset because this data spanning over five decades, this dataset offers a comprehensive view of automotive history.

II. LITERATURE REVIEW

The author discusses the application of Artificial Neural Networks (ANN) [3][4] in predicting share market prices, highlighting the challenges faced in predicting the chaotic nature of share prices and the suitability of ANN in identifying hidden patterns for accurate predictions. The authors emphasize the limitations of traditional methods such as technical and fundamental analysis and advocate for the use of ANN due to its ability to learn from training and experience. The paper outlines the system architecture, including the training and prediction phases, as well as the specific inputs used for training the network, such as General Index, Net Asset Value, P/E ratio, Earnings per Share, and Share Volume. The authors also delve into the Back propagation algorithm, detailing its role in adjusting connection weights and membership functions during the training sessions.

The author proposed the application of Artificial Neural Network (ANN) [2][13] as a prediction technique to improve accuracy in modelling with less dependency on experimental data, particularly in the context of machining processes. It explores the use of ANN for predicting output parameters to optimize machining parameters, aiding in process planning and parameter optimization. The paper delves into the concept of ANN as an interconnected group of nodes inspired by the human brain's neural network, capable of machine learning and pattern recognition. The paper emphasizes the advantages of ANN, such as its ability to continue functioning even if a component fails adaptability to various applications, and the capacity to learn without the need for reprogramming. It also highlights the challenges in applying ANN to machining processes, such as the need for proper training and the requirement for emulating the architecture of a neural network, which entails high processing time for large neural networks.

The authors in [10] and [11] both focus on the application of artificial neural networks (ANN) in predicting the price of second-hand cars and the travel time of buses, respectively. The paper’s authors aims to assess the feasibility of predicting the price of second-hand cars using ANN. The study collected data for 200 cars from various sources and applied four different machine learning algorithms. The results indicated that support vector machine regression produced slightly better results than using a neural network or linear regression. However, the study suggests that further investigations with a larger dataset and more experimentation with different network types and structures are required to obtain better predictions. Both papers highlight the potential of ANN in predicting various transportation-related factors, such as the price of second-hand cars and bus travel time, and emphasize the importance of accurate predictions for improving transportation systems and user satisfaction.

III. PROPOSED METHODOLOGY

Prediction method analysis

An open source machine learning and data science platform that provides data and notebooks for data scientists and data analysts is the source of the used car pricing dataset, which is chosen with the necessary features and parameters. Using machine learning techniques, the necessary data is cleaned and pre-processed before any price prediction algorithms are applied. Next, we must apply train test split to keep the data divided into two sections for training and validation using train and test data, respectively, after pre-processing and cleaning the data. The next step is to use a multilayer neural network model and use the r square score to forecast the test and train accuracy results. We have selected 15 random samples from test data and plotted actual and predicted price. We can also use Random Forest machine learning model, but it accuracy prediction r square score much lower than multilayer neural network model.

Artificial intelligence finds the machine learning technique appealing since it is predicated on the idea of learning by experience and training. In machine learning, connection weights can be modified to enhance network performance, making connectionist models like artificial neural networks (ANNs) a suitable fit.

Our system architecture

The neural network approach is appropriate for this type of chaotic system as it does not require us to comprehend the answer. One significant benefit of neural network techniques is this [9]. However, with traditional methods, we have to have a thorough understanding of the inputs, algorithms, and outcomes. All that is required of the neural network is to display the appropriate output for the supplied inputs. The network will imitate the function after receiving enough training [9], [10]. The network will learn to disregard any inputs that don't add to the output during the tanning process, which is another benefit of neural networks [9], [10].

In the training phase of our system, the back propagation algorithm is employed to find certain parameters called weights from this section. The same equations employed in the training phase are applied to these weights in the prediction phase. This is the fundamental design of our system; it is referred to as a feed forward network. Nine automotive attributes were used as input for the feed forward network. Due to category attributes and variables creating several columns during pre-processing, the actual input layer size was 215.

Back propagation with Feed foreword NN

The process of back-propagating errors from the output layers towards the input layer during training sessions is known as the back-propagation algorithm [8], [11],[14]. Since the hidden units can't be trained using target values, they must be trained using the mistakes from the earlier layers, which calls for back-propagation. A target value is present in the output layer and is used for comparison with the calculated value. The connection weights are changed in real time when the mistakes spread back across the nodes. Training will continue until the weight inaccuracies are minimal enough to be tolerated. Conversely, the Backpropagation Algorithm has a computational complexity of only O(n).[18],[19] The primary requirements for making an effective share price prediction are these algorithmic elements.

The main steps using the Back propagation algorithm as follows:

Step - 1: Calculate the relevant output by feeding the normalized input data sample;

Step - 2: Determine the discrepancy between the target(s) and the output(s);

Step - 3: Membership functions and connection weights are modified;

Step - 4: IF Error>Tolerance THEN go to Step1ELSE

Step - 5: Stop

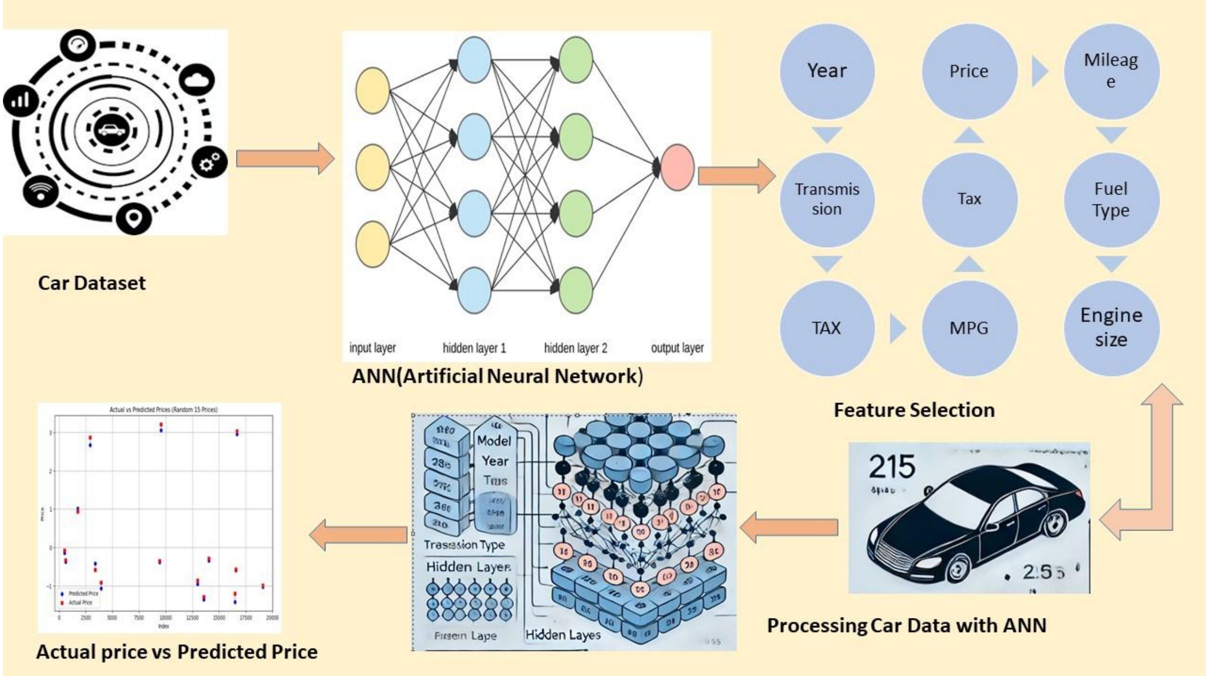


Figure 1: Proposed Model.

IV. SYSTEM IMPLEMENTATION

A. Model Implementation and Training

The neural network was implemented using Python's TensorFlow library. The training process involved feeding the training dataset into the network, adjusting the weights using backpropagation, and evaluating the model's performance using the test dataset. The network was trained for 100 epochs, with each epoch representing a complete pass through the training data.

Code snippet to create neural network model:

```
model = Sequential()

model.add (Dense(1000, activation='relu'))

model.add (Dense(800, activation='relu'))

model.add (Dense(600, activation='relu'))

model.add (Dense(300, activation='relu'))

model.add (Dense(1)) # O/P

model.compile (optimizer='rmsprop', loss='mse')
```

Table 1: Neural Network Model

Layer(type)	Output Shape
dense (Dense)	(None, 1000)
dense_1 (Dense)	(None, 800)
dense_2 (Dense)	(None, 600)
dense_3 (Dense)	(None, 300)
dense_4 (Dense)	(None, 1)

Total params: 3,356,004 (12.80 MB)

Trainable params: 1,678,001 (6.40 MB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 1,678,003 (6.40 MB)

Training Phase

The price of a new car cannot be directly predicted using a neural network model once it has been developed or defined. We must use a dataset to fine-tune this model in order to use it for price prediction. Setting the initial random weights for each node across all layes in a neural network model is known as tuning. Neural network tuning is divided into two stages: the training phase and the prediction/testing phase. Before fine-tuning, the dataset is divided into two sections: the training dataset and the testing component. The propagation phase and the weight update phase are two further divisions of the training phase..

The data from the input training dataset is first normalized using the following formula in the propagation phase 1 in order to feed the network into the input nodes:

$$= \frac{v - minA}{maxA - minA}(new_{maxA} - new_{minA}) + new_{minA} \quad \ldots [1]$$

Here,

V = Actual Input.

Min A, Max A = Boundary values of the old data range.

New min A, New max A = Boundary values of the new data range.

Because the back propagation can only handle data between minus one to one, in this example, it is -1 and 1. [12]

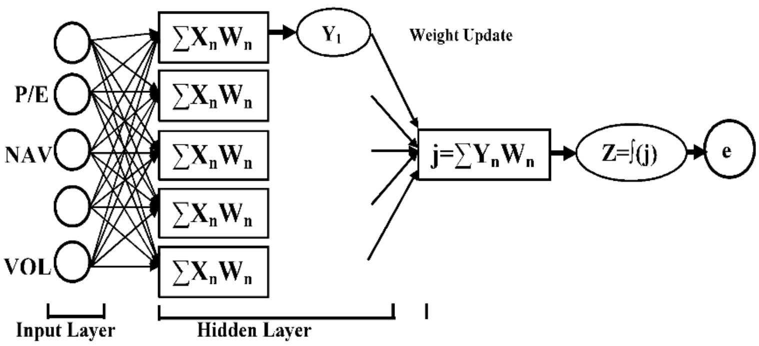


Figure 2. Training phase

Figure 2 illustrates how normalized input data are fed into the input layer from the left side. Each input data set is then multiplied by the weights before entering the hidden layer neurons. Figure 2 also describes the function of a single neuron; in our model, we utilized a single hidden layer. The hidden layer neurons in our model perform the same tasks as the neurons in the input layer. Subsequently, every neuron transfers its output to the subsequent neuron in the output layer. Similar to the hidden layer neuron, the output layer calculates and produces a final output that is compared to the real output to determine the error signal (e).

$$e = \frac{actual\ output - ANN\ output}{actual\ output} \quad \ldots [2]$$

The following formula is used to update the weight using the error (e) that is generated during the Propagation Phase:

Updated Weight= weight (old) + learning rate * output error * output (neurons i) * output (neurons i+1) * (1-

Output (neurons i+1))   …..[3]

To update weight, the aforementioned procedure is carried out in each weight matrix inside the network. Until the sum of square error is zero or very near to zero, the Phase 1 and Phase 2 procedures are applied again. Similar to the first illustration, every neuron consists of two units. The input signals and weights coefficients are added in the first unit [12]. Next, this output enters the neuron's second unit, which houses the nonlinear activation function (in our model, the relu function serves as the activation function) [13].

The prediction phase

After training, the model was evaluated on the test set to assess its accuracy. The primary metric used for evaluation was the R-squared score, which indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. The model achieved an R-squared score of 0.96 on the training set and 0.95 on the test set, indicating a strong correlation between predicted and actual prices.

The difference in R-squared scores between the training and test sets was minimal, demonstrating that the model did not suffer from overfitting. Additionally, the loss values for both training and validation sets decreased steadily with each epoch, further confirming the model's robustness.

Original dataset Analysis & Cleaning

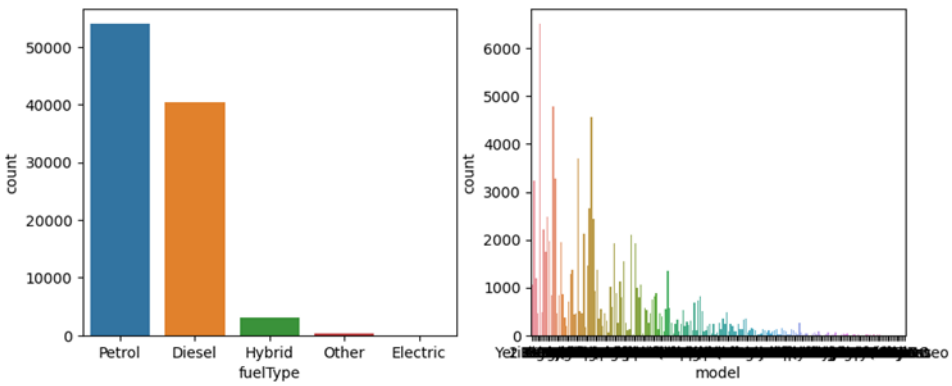
We have excel file with 90000 plus records files which contains information about various features of new manufactured car from 1970 to 2023 from various manufacturer worldwide. We have used various python library for cleaning of this dataset and define & train the ANN model to predict price of new car.

Table 2: Actual Dataset

No	Model	Year	Price	Transmission	Mileage	Fuel Type	Tax	mpg	Engine Size	Manufacturer
0	I10	2017	7495	Manual	11630	Petrol	145	60.1	1.0	Hyundai
1	Polo	2017	10989	Manual	9200	Petrol	145	58.9	1.0	Volkswagen
2	2 Series	2019	27990	Semi-Auto	1614	Diesel	145	49.6	2.0	BMW
3	Yeti Outdoor	2017	12495	Manual	30960	Diesel	150	62.8	2.0	Skoda
4	Hesta	2017	7999	Manual	19353	Petrol	125	54.3	1.2	Ford

df.info()

There is no null value or missing value records in dataset, so all records are useful to train our model.



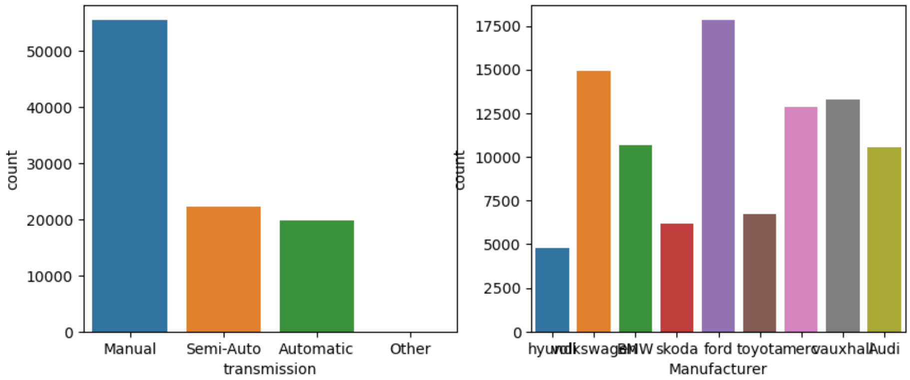


Figure 3. Categorical columns each value versus records count

Above graphs represent each category related records count for each categorical field. There is four categorical Fields so we have four graphs.

Each cell of above diagram indicates labeled fields correlation. Fields which have more correlation, either positive or negative with price (our estimation field) helps to predict price more accurately.

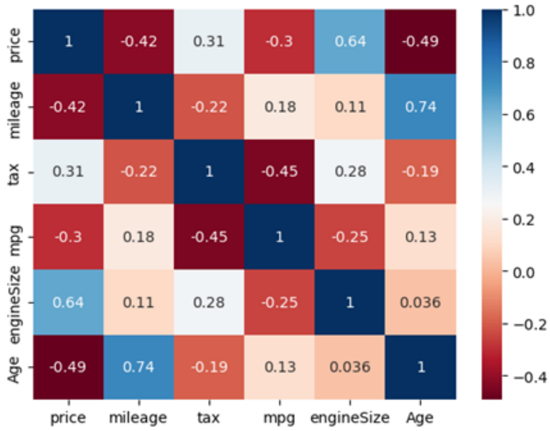


Figure 4. Correlation of all columns of Dataset

V. RESULTS AND DISCUSSION

Ninety thousand new car records have been split up, with 80% of the records serving as training records and 20% serving as test records. Using training records, we have trained the neural network model mentioned above. For the neural network's training, we employed 100 epochs and 100 sample batches. When we use a batch size of 100, we submit 100 samples to the input layer in a single cycle and record the result for each sample. Next, we contrast the sample price generated for each car with the sample actual price. One sample error separates these two values from one another. As part of the back propagation process, the neural network will find the average of these 100 sample mistakes and update the weight of its nodes based on that average value. Once all 80% of the 90,000 samples have been used, the process will be repeated for the remaining 100 samples, marking the end of an era. Next, we analyse the difference between the neural network's anticipated and real prices for a 20% test sample of new cars. Once all test samples have been supplied to the neural network, enter the subsequent test sample and measure the error. Next, we determine the average error value across all test samples; this one average error value is referred to as the validation loss of the first epoch (round). The same procedure is then carried out once more using 80% training samples and 20% test samples of brand-new automobiles. Consequently, we obtain the validation loss value of the second epoch (round) once more. The purpose of performing these 100 epochs is to minimize validation loss. We may state that the neural network is trained and ready to forecast the price of any new car based on its attributes after 100 epochs.

We transformed categorical attributes into numerous columns during the pre-processing stage of the data based on the attributes' multiple category values.

As a result, our 10-column dataset has been transformed into a 215-column dataset. The first dense hidden layer in the ANN model above has 1000 nodes. We then reduce the number of nodes in the successive layers such that an ANN can solve our nonlinear problem of predicting the price of a new car. While multilayer ANNs can depict complicated nonlinear relationships between input features and prediction attributes, single layer ANNs can only tackle linear problems. In our instance model, pricing is a prediction attribute with a complex nonlinear connection, and input data include things like mileage, fuel type, capacity, etc. As we wish to anticipate a single price value for every new car sample, the output layer has a single node.

Validation loss and training loss decreases as number of epoch increases.

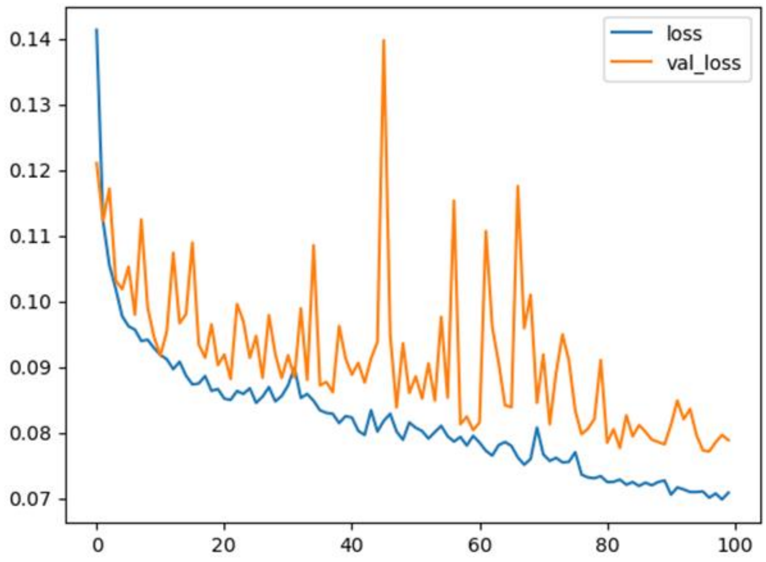


Figure 5: Loss vs. Epoch

Above Graph shows validation loss and training loss versus epoch counts. Both type losses are decreases as number of epoch increases.

As was indicated during the training phase, we test the ANN model using 20% of the test samples from new cars after each training cycle to get the validation loss for that round (epoch). For our ANN to be trained with the best accuracy and minimal validation loss, we have set 100 epochs. The plot of epoch against validation loss and loss is shown in Figure 5. When testing fresh automobile samples are input into an ANN, the average error is known as validation loss, and when training sample input is averaged, it is known as loss. As the number of epochs increases, we can see that both losses decrease. Following 100 epochs, the accuracy is 96% and the loss value is 0.04, or 4%. r - Square value is an additional ANN model accuracy metric. The relationship between the real and expected prices of the sample of cars is indicated by the R square value. Therefore, a closer value for this parameter suggests that our prediction model is ideal, and for both training and test samples, our r square accuracy is 0.96 (96%). Thus, we can conclude that our model is fully trained and prepared for use in predicting the price of any new car.

Difference between two scores: 0.017

We took 15 samples from the testing samples of new cars using a random function, which we then fed into an artificial neural network (ANN) to forecast the price. The resulting graph shows the number of samples versus the sample real price (blue points) and sample anticipated price (red points). Ideally, every red 15 point and every blue 15 point would be in the same spot, one above the other, but our ANN model's 4% loss makes this impossible. For fifteen randomly chosen samples, the graph shows the actual and expected prices. Nearly fifteen of the projected price points match or are close to the corresponding real pricing values. Our trained artificial neural network is 96% accurate, as demonstrated by this visual graph..

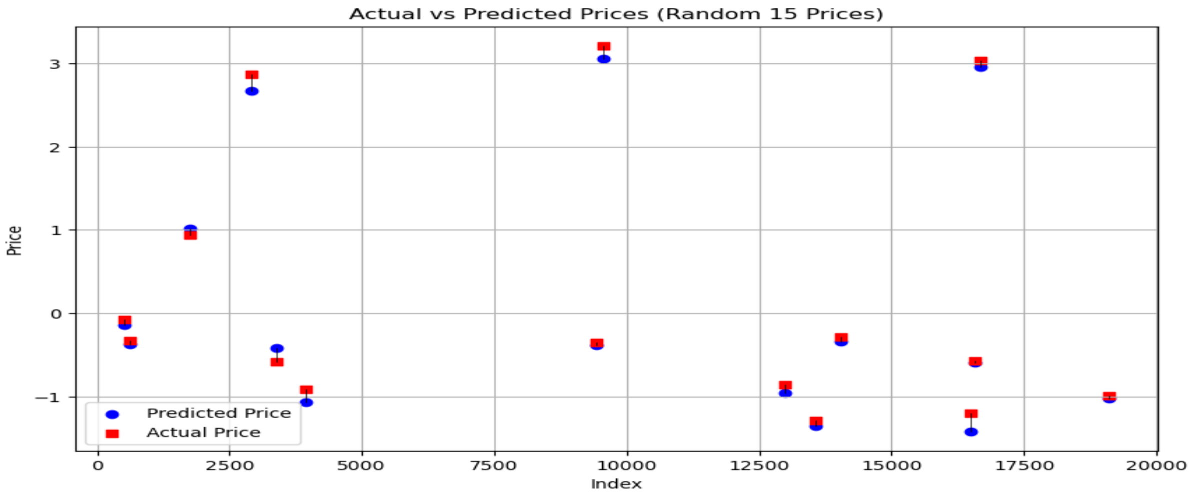


Figure 6: Points of Actual and Predicted Price

1. Square value indicates association between predicted value and actual value of test data and train data. So we observed that almost 96% accuracy is achieved in prediction of car price

2. Fig.5 Graph indicates actual price and predicted price for 15 randomly selected samples. Almost 15 all points of predicted price are near or coincide with corresponding actual price points.

VI. CONCLUSION

This research demonstrated the effectiveness of Artificial Neural Networks (ANNs) in predicting car prices based on various features. The ANN model developed in this study achieved high accuracy, making it a valuable tool for both manufacturers and consumers. Manufacturers can use this model to set competitive prices for new car models, while consumers can use it to verify the fairness of car prices in the market.

Future work could explore the integration of additional features, such as real-time market data, to further enhance the model's predictive capabilities. Additionally, experimenting with different neural network architectures and hyper parameters could yield even better results.

Overall, the use of ANNs in car price prediction represents a significant advancement in the application of AI to real-world problems, offering practical benefits to both industry professionals and end-users.

REFERENCES

[1] Z.Tang and P.A.Fishwick, "Backpropagation neural nets as models for time series forcasting," ORSA journal on computing, vol.5, No. 4, pp 374-384, 1993.

[2] J.H.wang and J.Y.Leu, "stock market trend prediction using ARIMA-based neural network,"Proc. Of IEEE conference on neural networks, vol.4, pp.2160-2165, 1996

[3] Kimoto,T., asakawa, K., Yoda , M,. and Takeoka, M. (1990),Stock market prediction system with modular neural network, in proceedings of the International Joint Conference on Neural Network, 1-6.

[4] Mizuno, H., Kosaka , M., Yajima , H. and Komoda N. (1998), Application of Neural Network to Technical Analysis of Stock Market Prediction, Studies in Information and Control , vol.7, no.3, pp.111-120.

[5] Sexton, R. S., R. E. Dorsey and J.D.Dohnson (1998) , Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation, Decision Support systems 22, 171- 185.

[6] Phua, P.K.H. Ming, D., Lin, W. (2000), Neural network With Genetic Algorithms For Stocks Predictio, Fifth Conference of the Association of Asian-Pacific Operations Research Societies, 5^{th} – 7^{th} july, Singapore.

[7] Samarth Agarwal,Manol Jindal,G.N.Pillai "Momentum Analysis based Stock Market Prediction using ANFIS". In Proceeding of the International Multiconference of Engineering and Scientists 2010 Vol.1, IMECS 2010, March 2010, Hong Kong.

[8] Rumelhart, D.D.m Hinton, G.E. and Williams, R.J., Learning Internal Representation, Man, and

Cybernetics (SMC"91), 1991. 1913-1918.
[9] "Feedforward neural Networks: An Introduction" by Simo Haykin page (2-4)
[10] "Artificial Intelligence a morden approach" (second edition) by Stuart Russell, Peter Norvig 2004
[11] Simon Haykin, "Neural Network A Comprehensive Foundation", second edition, Prentice Hall, 1998, page 161 – 173.
[12] Robert J. Van Eyden. "The Application of Neural Networks in the Forcasting of Share Prices". Finance and Technology Publishing, 1996.
[13] W.Duch and N. Jankowski, "Transfer functions: hidden possibilities for better neural networks.", 9th European Symposium on Artificial Neural Networks (ESANN), Brugge 2001. De-facto publications.
[14] Y.-Q. Zhang and A. Kandel, "Compensatory Genetic Fuzzy Neural Networks and Their Applications," Series in Machine Perception Artificial Intelligence, Volume 30, World Scientific, 1998.
[15] Peerun, Saamiyah, Nushrah Henna Chummun, and Sameerchand Pudaruth. "Predicting the price of second-hand cars using artificial neural networks." The Second International Conference on Data Mining, Internet Computing, and Big Data. No. August. 2015.
[16] Mhatre, Mitali S., et al. "A review paper on artificial neural network: a prediction technique." International Journal of Scientific & Engineering Research 6.12 (2015): 161-163.
[17] Mohamed, Zahraa E. "Using the artificial neural networks for prediction and validating solar radiation." Journal of the Egyptian Mathematical Society 27 (2019): 1-13.
[18] Amita, Johar, S. S. Jain, and Pradeep Kumar Garg. "Prediction of bus travel time using ANN: a case study in Delhi." Transportation Research Procedia 17 (2016): 263-272.
[19] Khan, Zabir Haider, Tasnim Sharmin Alin, and Md Akter Hussain. "Price prediction of share market using artificial neural network (ANN)." International Journal of Computer Applications 22.2 (2011): 42-47.