Journal of Engineering, Electrical and Informatics
Vol.3, No.3 oktober 2023
e-ISSN: 2809-8706; p-ISSN : 2810-0557, page 34-44
DOI: https://doi.org/10.55606/jeei.v3i3.2254

The Implementation of Artificial Neural Networks for Stock Price Prediction

Akbar Maulana
Faculty of Science & Technology, Yogyakarta Technology University

Enny Itje Sela
Faculty of Science & Technology, Yogyakarta Technology University

Corresponding author: akbarmaulana55695@gmail.com

Abstract. *This research is based on a problem that is difficult to predict stock prices, especially for beginners. Stock prices are hard to predict because they are fluctuating. Users will be easier to predict stock prices through artificial neural networks using Multilayer Perceptron. This MLP is a variant of an artificial neural network and is a development of perceptron. The selection of the Multilayer Perceptron method is based on the ability to solve various problems both classification and regression. The research conducted by the author is a regression problem as the MLP is tasked to predict the close price or closing price of stock after seven days. The results of the model built are able to predict stock prices and produce good accuracy because the resulting RMSE value produced 0.042649862994352014, which is close to 0.*

Keywords: *Machine Learning, Stock Price Prediction, Neural Network, Multilayer Perceptron, MLP.*

INTRODUCTION

Shares or stocks are evidence of value that shows ownership in a company (Bhattacharjee et al., 2021; Gibbs et al., 2007). Buying shares in a company can mean that we have ownership of that company. Besides, it has advantages in both the short and long term. Short-term profits can be obtained when share or stock buyers purchase the stock when its prices are low and sell them when the prices are rising or in an upward trend (capital gain). The long-term advantage of owning shares is that the owner gets a stock of the company's profits, but the period is relatively long (Huang & Fang, 2021).

Behind these profits, stock prices also have large risks because its prices are very fluctuating. This condition can be influenced by supply and demand, inflation, interest rates, as well as company performance. Therefore, someone who wants to or has just tried the stock market often finds it difficult to predict stock prices due to a lack of knowledge and experiences (Afifah & Fauziyyah, 2023; Yang & Zhou, 2016).

Received August 30, 2023; Revised September 30, 2023; Accepted October 31, 2023
* Akbar Maulana, *akbarmaulana55695@gmail.com*

A fairly large risk arises when someone buys a share without first analyzing the stock value. The risks are the dissolution of the company (liquidation) and falling stock prices. As a result, someone is required to analyze the value of shares first before purchasing stocks to get maximum profits and minimize losses (Dewan Standar Akuntansi Keuangan IAI, 2018; Husnul et al., 2017; Jaggi et al., 2021; Joseph et al., 2023; Napitupulu, 2021).

Several methods can be used to predict stock prices. One of these methods is Machine Learning. It is a sub of Artificial Intelligence (AI), which aims to improve knowledge and performance (Provost & Fawcett, 2013). Machine Learning also has various algorithms. One of these algorithms is Multilayer Perceptron, which is a development of the perceptron (Abbahaddou et al., 2022; Awad et al., 2021; Polamuri et al., 2019; Satria, 2023; Sheth & Shah, 2023).

Some research to predict stock prices has been carried out in previous years. Research conducted by Rasheed Khaled (2012) used Genetic Algorithms and Strategy Algorithms. In this research, the Genetic Algorithm produced an accuracy rate of 73.78%, while the Evolutionary Strategy Algorithm produced an accuracy rate of 71.77%. Research conducted by M Abdul Dwiyanto Suyudi, Esmeralda C. Djamal, Asri Maspupah (2019) showed that stock price predictions using LSTM showed poor results with accuracy for training data of 94.16% and test data of 55.26%. These results were obtained using Adam optimization with a learning rate of 0.001 and an epoch of 200.

The research above can be concluded that LSTM is considered to have poor performance in predicting stock prices. Therefore, the author tries to apply the Multilayer Perceptron method to predict stock prices. The author tries to apply the Multilayer Perceptron (MLP) backpropagation algorithm as MLP is a machine learning algorithm that can solve regression problems in the form of sequential data.

METHOD

The framework for this research is started with the problems faced by users who have difficulty in predicting stock prices (Sugiyono, 2019). Based on the background explanation, artificial neural networks are able to predict stock prices with a fairly high level of accuracy so that neural networks can be used as a strategy to be applied in predicting stock prices. After going through the necessary processes, the resulting output is in the form of stock predictions, which is able to help users in the future.

The analysis data used the stock price of the Walt Disney Company (DIS), sourced from Yahoo Finance. These data were accessed using the yfinance library so the data did not need to be downloaded and could be accessed at any time as long as it is connected to the internet. The data was started from January 1, 2008 to April 14, 2023 and has 3847 rows. The features of this dataset include the date, open price, highest price, lowest price, close price, Adj Close, and Volume. An example data can be seen in Table 1 as follows:

Table 1. Example of Data Taken using Yfinance

No	Date	Open	High	Low	Close	Volume
1	2008-01-02	27,419	27,682	26,885	27,012	9269900
2	2008-01-03	27,046	27,165	26,859	26,953	9681100
3	2008-01-04	26,299	26,876	26,299	26,410	9550700
4	2008-01-07	26,622	26,715	26,223	26,435	10742900
5	2008-01-08	26,511	26,724	25,841	25,909	13014300

From these data, the column used in this research is only the close column so that the other columns are deleted. The data that will later be used for this research can be seen in Table 2:

Table 1Data After Column Deletion

No	Date	Close
1	2008-01-02	27,012
2	2008-01-03	26,953
3	2008-01-04	26,410
4	2008-01-07	26,435
5	2008-01-08	25,909

The Walt Disney stock data used was obtained by accessing Yahoo Finance using the yfinance library. This code aims to specify what stock data will be accessed. The Walt Disney stock data used in this research has a time span starting from January 1, 2008 to April 14, 2023.

The model architecture in the first stage is that the user searches for data to be used. The next stage is that the data will be applied to the designed system. After the data is processed in the designed system, the output that will be issued is a prediction of stock prices and a comparison with stock prices in the world.

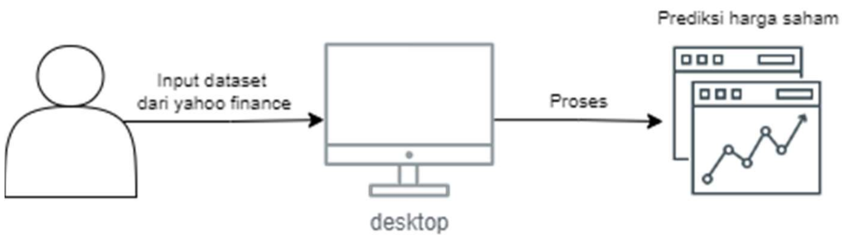

Figure 1Architecture Model

RESULTS AND DISCUSSION

The following is the implementation and discussion of the system created. The first stage is downloading and importing the required libraries. Next, access and download the Waltz Disney Company stock dataset on Yahoo Finance. After that, call the data into the dataframe and provide stock date parameters that will be included in the dataframe. Then, the dataframe will be read and displayed (Sadorsky, 2021; Sareen & Sharma, 2022; Satria, 2023).

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2008-01-02 00:00:00-05:00	27.419730	27.682730	26.885249	27.012506	9269900	0.0	0.0
1	2008-01-03 00:00:00-05:00	27.046440	27.165215	26.859797	26.953119	9681100	0.0	0.0
2	2008-01-04 00:00:00-05:00	26.299863	26.876764	26.299863	26.410152	9550700	0.0	0.0
3	2008-01-07 00:00:00-05:00	26.622250	26.715572	26.223510	26.435606	10742900	0.0	0.0
4	2008-01-08 00:00:00-05:00	26.511956	26.724052	25.841733	25.909605	13014300	0.0	0.0
...	...	...	...	...	...	...	...	...
3842	2023-04-06 00:00:00-04:00	99.440002	100.320000	98.550003	99.970001	7042500	0.0	0.0
3843	2023-04-10 00:00:00-04:00	99.300003	100.809998	98.900002	100.809998	8016600	0.0	0.0
3844	2023-04-11 00:00:00-04:00	101.160004	101.910004	100.290001	100.419998	7504600	0.0	0.0
3845	2023-04-12 00:00:00-04:00	101.250000	102.220001	97.699997	97.940002	9289200	0.0	0.0
3846	2023-04-13 00:00:00-04:00	98.510002	101.070000	98.510002	100.839996	8745000	0.0	0.0

3847 rows × 8 columns

Figure 2. Waltz Disney Stock Data

The next stage is the preprocessing stage. Of the many columns, only the close column will be used. After selecting only the close column, another preprocessing stage is carried out, by checking missing values, removing missing values, and normalizing the data by scaling the data using MinMaxScaler so that calculations in artificial neural network models can be easier. The formula for MinMaxScaler can be seen in equation 1:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

An example of a data scaling calculation can utilize the data used in this research. The data to be tested can be seen in Table 3:

Table 3. Example of the Data

Index	Close
0	27.01250267028809
1	26.9531192779541
2	26.41015243530273
3	26.435604095458984
4	25.90960502624512
5	25.587221145629883
6	26.019895553588867
7	25.72296524047852

From the dataset used, it is known that the largest value (max) and smallest value (min) are 201.910 and 13.424.

Then, the data is rearranged where the previous seven days' data will be used as input and the eighth day's data will be used as the target. The results of the above process can be seen in Figure 3:

```
Inputs:
        Day_1     Day_2     Day_3     Day_4     Day_5     Day_6     Day_7
0     0.072090  0.071775  0.068894  0.069030  0.066239  0.064528  0.066824
1     0.071775  0.068894  0.069030  0.066239  0.064528  0.066824  0.065249
2     0.068894  0.069030  0.066239  0.064528  0.066824  0.065249  0.065384
3     0.069030  0.066239  0.064528  0.066824  0.065249  0.065384  0.063133
4     0.066239  0.064528  0.066824  0.065249  0.065384  0.063133  0.063043
...        ...       ...       ...       ...       ...       ...       ...
3835  0.431839  0.442716  0.449241  0.460011  0.458048  0.457040  0.458844
3836  0.442716  0.449241  0.460011  0.458048  0.457040  0.458844  0.459163
3837  0.449241  0.460011  0.458048  0.457040  0.458844  0.459163  0.463619
3838  0.460011  0.458048  0.457040  0.458844  0.459163  0.463619  0.461550
3839  0.458048  0.457040  0.458844  0.459163  0.463619  0.461550  0.448392

[3840 rows x 7 columns]
Targets:
        Target
0     0.065249
1     0.065384
2     0.063133
3     0.063043
4     0.058272
...        ...
3835  0.459163
3836  0.463619
3837  0.461550
3838  0.448392
3839  0.463778

[3840 rows x 1 columns]
(3840, 7)
(3840, 1)
```

Figure 3. Results After Data Reordering

After the data is rearranged, the data will be divided into three, namely training data, validation data and test data with a comparison ratio of 70%:15%:15%. The code for data sharing can be seen in Figure 4:

```
# Split the data into inputs and targets
inputs = inputs_df[:-1]  # Inputs: all rows except the last one
targets = targets_df[1:]  # Targets: all rows except the first one

train_index = int(0.70 * len(inputs))
val_index = int(0.85 * len(inputs))
#usually 70:15:15

# Split the data into training, validation, and testing sets
train_inputs, train_targets = inputs[:train_index], targets[:train_index]
val_inputs, val_targets = inputs[train_index:val_index], targets[train_index:val_index]
test_inputs, test_targets = inputs[val_index:], targets[val_index:]

# Print the sizes of the training, validation, and testing sets
print(f"Train set size: {len(train_inputs)}")
print(f"Validation set size: {len(val_inputs)}")
print(f"Test set size: {len(test_inputs)}")
```

```
Train set size: 2687
Validation set size: 576
Test set size: 576
```

Figure 4 Dataset Division Code

The next stage is model creation. The model built in this research has seven neurons as the input layer, two hidden layers, each hidden layer having seven neurons, and an output layer consisting of one neuron. The architecture of the research used can be seen in Figure 5:

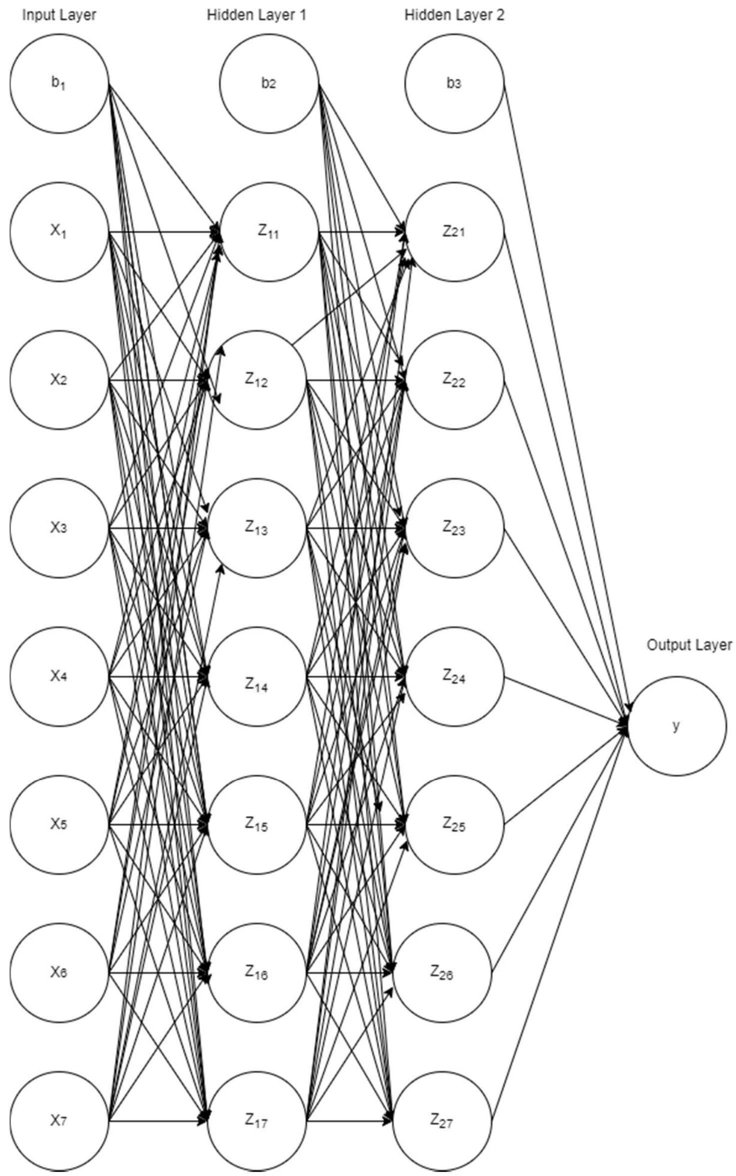

Figure 5. Architecture Model

Next is compiling the model. In this research, the optimizer used is *stochastic gradient descent*. For the activation function process, there are three options, namely *tanh*, *relu*, and *sigmoid*. Sliding windows were also added in this research with the aim of improving model performance. Sliding windows work by dividing data into windows of a fixed size that overlap each other. The artificial neural network is then trained to predict targets based on input values from sliding windows.

The trained model will be validated using validation data that have been prepared during the data sharing process. After that, the model of the artificial neural network will be saved and re-used during the testing stage. At this stage, if the loss from the training and validation process is still quite high, then parameters, such as learning rate, epoch, activation function, size of sliding windows, and so on will be readjusted.

The next stage is implementing the test data to the model that has been saved. After testing process, the data that have been scaled using MinMaxScaler will be inverted or returned to its original value (Deepa & Ramesh, 2022; Shaheen et al., 2020). The model evaluation in this research used RMSE. The output or prediction output from the trained model will be reduced to the original data and then averaged and rooted. Next process is making graphs to see the comparison between the predicted results and the original data (ground truth). The next stage is to compare the predicted value with actual stock data.

There were several trials to replace more suitable parameters in the model being built. These parameters are learning rate, epoch, batch size, window size, and activation function. The experimental results can be seen in the Table 4. The smallest RMSE results obtained are 0.042649862994352014 and it can be stated that the model created has a good performance.

Table 4. Results of Model Testing Experiments

Learning Rate	Epoch	Batch Sizes	Windows Sizes	Seed	Activation	RMSE
0.05	1000	42	4	42	Relu	0.042649862994352014
0.01	1000	21	4	42	Relu	0.22088035881599877
0.01	1000	42	6	21	Relu	0.060893733828284655
0.08	1000	42	4	42	Relu	0.16104509002174425
0.01	1000	42	6	21	Tanh	0.13048368913533623
0.05	1000	42	4	42	Tanh	0.15935786586165798
0.01	1000	21	4	21	Tanh	0.13400511143257018
0.01	1000	21	4	21	Sigmoid	0.17954298277176617
0.05	1000	42	4	42	Sigmoid	0.16104509002174425
0.01	1000	42	6	21	Sigmoid	0.17875569186199866

Although the resulting RMSE is considered good because it is below 0.1, there is still quite a large difference between the predicted stock price and the actual stock price. However, this only happens at the beginning. This can be seen through the graphs in Figure 6 and in Figure 7:

Figure 6. Graph

	Actual	Prediction	Difference
0	177.300003	164.145940	13.154063
1	181.169998	165.037331	16.132667
2	181.179993	169.223763	11.956229
3	177.679993	169.597704	8.082289
4	178.440002	172.710021	5.729981
5	179.119995	170.545106	8.574889
6	178.580002	167.024665	11.555336
7	178.690002	170.972408	7.717594
8	179.089996	167.994247	11.095750
9	175.990005	166.251788	9.738218
10	176.119995	168.859578	7.260417
11	173.429993	167.433283	5.996710
12	171.440002	164.966429	6.473573
13	172.259995	165.291836	6.968159
14	173.639999	161.809490	11.830509
15	171.279999	160.821701	10.458298
16	172.779999	163.426262	9.353736
17	171.889999	161.330102	10.559898
18	169.559998	160.952404	8.607593
19	163.029999	164.130189	-1.100190

Figure 7. Comparison of Predicted Prices and Original Prices

CONCLUSIONS AND RECOMMENDATIONS

From the research that has been conducted, it can be concluded that stock price prediction using a multilayer perceptron artificial neural network with backpropagation has good performance. This is evidenced by the RMSE value, which is relatively small with an RMSE value of 0.042649862994352014. Another supporting evidence is the price difference between the predicted price and the actual price. However, several problems were still found because the predicted price and the actual price still had quite a large difference at the beginning of the test. Future researches are expected to change the parameters in this test, for example changing the optimizer, epoch, batch size, learning rate, and others.

REFERENCE

Abbahaddou, K., Chiadmi, M. S., & Aboulaich, R. (2022). An Enhanced Adaptative System based on Machine Learning for Predicting the Evolution of Islamic Stock Prices. *WSEAS Transactions on Business and Economics*, *19*. https://doi.org/10.37394/23207.2022.19.150

Afifah, S. N., & Fauziyyah, N. (2023). Dampak Resesi 2023 terhadap Harga Saham di Indonesia. *MIZANIA: Jurnal Ekonomi Dan Akuntansi*, *3*(1). https://doi.org/10.47776/mizania.v3i1.612

Awad, A. L., Elkafas, S. M., & Fakhr, M. W. (2021). ROle Of Machine Learning In Predicting Stock Prices: A Literature Survey. *Journal of Management Information and Decision Sciences*, *24*(SpecialIssue1).

Bhattacharjee, J., Singh, R., & Kajol, K. (2021). Risk perception in respect of equity shares: A literature review and future research agenda. In *DLSU Business and Economics Review* (Vol. 30, Issue 2).

Deepa, B., & Ramesh, K. (2022). Epileptic seizure detection using deep learning through min max scaler normalization. *International Journal of Health Sciences*. https://doi.org/10.53730/ijhs.v6ns1.7801

Dewan Standar Akuntansi Keuangan IAI. (2018). *Standar Akuntansi Keuangan*. 934.

Gibbs, H. K., Brown, S., Niles, J. O., & Foley, J. A. (2007). Monitoring and estimating tropical forest carbon stocks: Making REDD a reality. *Environmental Research Letters*, *2*(4). https://doi.org/10.1088/1748-9326/2/4/045023

Huang, B., & Fang, X. (2021). Market Sentiment, Valuation Heterogeneity, and Corporate Investment: Evidence from China's A-Share Stock Market. *Emerging Markets Finance and Trade*, *57*(8). https://doi.org/10.1080/1540496X.2019.1672531

Husnul, H. M., Hidayat, R. R., & Sulasmiyati, S. (2017). Analisis pengaruh inflasi, kurs (IDR/USD), Produk Domestik Bruto dan harga emas dunia terhadap Indeks Harga Saham Gabungan. *Jurnal Administrasi Bisnis*, *53*(1).

Jaggi, M., Mandal, P., Narang, S., Naseem, U., & Khushi, M. (2021). Text mining of stocktwits data for predicting stock prices. *Applied System Innovation*, *4*(1). https://doi.org/10.3390/asi4010013

Joseph, S., Mduma, N., & Nyambo, D. (2023). A Deep Learning Model for Predicting Stock Prices in Tanzania. *Engineering, Technology & Applied Science Research*, *13*(2). https://doi.org/10.48084/etasr.5710

Napitupulu, R. D. (2021). PENGARUH CR DAN HARGA SAHAM TERHADAP ROA PADA PERUSAHAAN SUB SEKTOR FOOD AND BEVERAGE YANG TERDAFTAR DI BURSA EFEK INDONESIA. *Journal of Information System, Applied, Management, Accounting and Research*, *5*(1). https://doi.org/10.52362/jisamar.v5i1.361

Polamuri, S. R., Srinivas, K., & Krishna Mohan, A. (2019). Stock market prices prediction using random forest and extra tree regression. *International Journal of Recent Technology and Engineering*, *8*(3). https://doi.org/10.35940/ijrte.C4314.098319

Sadorsky, P. (2021). A Random Forests Approach to Predicting Clean Energy Stock Prices. *Journal of Risk and Financial Management*, *14*(2). https://doi.org/10.3390/jrfm14020048

Sareen, A., & Sharma, S. (2022). Assessing Financial Distress and Predicting Stock Prices of Automotive Sector: Robustness of Altman Z-score. *Vision*, *26*(1). https://doi.org/10.1177/0972262921990923

Satria, D. (2023). PREDICTING BANKING STOCK PRICES USING RNN, LSTM, AND GRU APPROACH. *Applied Computer Science*, *19*(1). https://doi.org/10.35784/acs-2023-06

Shaheen, H., Agarwal, S., & Ranjan, P. (2020). Minmaxscaler binary pso for feature selection. *Advances in Intelligent Systems and Computing*, *1045*. https://doi.org/10.1007/978-981-15-0029-9_55

Sheth, D., & Shah, M. (2023). Predicting stock market using machine learning: best and accurate way to know future stock prices. In *International Journal of System Assurance Engineering and Management* (Vol. 14, Issue 1). https://doi.org/10.1007/s13198-022-01811-1

Sugiyono. (2019). Metode penelitian dan pengembangan (research and development/R&D). *Alfabeta*.

Yang, C., & Zhou, L. (2016). Individual stock crowded trades, individual stock investor sentiment and excess returns. *North American Journal of Economics and Finance*, *38*. https://doi.org/10.1016/j.najef.2016.06.001