MECOMARE, 11 (2) (2022) pp. 100-108

Published by: TRIGIN Publisher

International Journal of Mechanical Computational and Manufacturing Research

Journal homepage: www.trigin.pelnus.ac.id

Mushroom Production Prediction Model using Conjugate Gradient Algorithm

Yosua Chandra Simamora[1], Solikhun[2], Lise Pujiastuti[3], Mochamad Wahyudi[4]

[1]STIKOM Tunas Bangsa Pematangsiantar, Indonesia

[2]AMIK & STIKOM Tunas Bangsa, Indonesia

[3]STMIK Antar Bangsa, Indonesia

[4]Bina Sarana Informatika University, Indonesia

ARTICLEINFO

Article history:

Received Jun 11, 2022
Revised July 09, 2022
Accepted August 28, 2022

Keywords:

ANN
Conjugate Gradient
Prediction
Mushroom Plant Production

ABSTRACT

Mushrooms are heterotrophic living things that act as saprophytes on dead plants. Mushrooms contain many important substances such as protein, amino acids, lysine, histidine, etc. Mushrooms tend to be better consumed than animal meat, even the content of lysine and histidine contained in mushrooms is greater than eggs. In recent years the volume of Mushroom Demand has increased, while production has decreased, especially on the island of Sumatra, namely in 2020 and 2021. Therefore, it is necessary to predict the estimated production of mushroom plants on the island of Sumatra so that the government on the island of Sumatra has clear data references to determine policies and make the right steps so that the production of mushroom plants on the island of Sumatra does not continue to decline. The method used in predicting is one of the ANN methods, namely the Conjugate Gradient Algorithm. The data used in this paper is Vegetable Crop Production data from 2014-2021 which was obtained from the website of the Central Statistics Agency. Based on this data, network architecture models such as 3-10-1, 3-15-1, 3-20-1, 3-25-1, 3-30-1, will be formed and defined. From the five models, training and testing values were obtained which showed that the most optimal architectural model was 3-10-1 with a Performance/MSE test value of 0.00055034. This value is the smallest of the 5 architectural models after the training and testing process. From this it can be concluded that this model can be applied to predict mushroom production on the island of Sumatra.

Corresponding Author:

Mochamad Wahyudi
Bina Sarana Informatics University
Jl. Kramat Raya No. 98, DKI Jakarta, 10450, Indonesia
Email: wahyudi@bsi.ac.id

1. INTRODUCTION

Mushrooms are agricultural products that have a high protein content. The USDA says there are 3.1 grams of protein in 100 grams of mushrooms (Pujiati, 2018). Besides that, it tastes delicious. Mushrooms also play an important role in maintaining body fitness, and are useful for fighting disease (Savitri, 2016). The mushroom business potential is currently seen as very large and can open up business opportunities (Waris & Khasanah, 2019). Mushrooms that can be consumed and have a large market are stromal mushrooms, oyster mushrooms, ear mushrooms, shitake mushrooms, and button mushrooms (Lianah, 2020). The Deputy Director of Vegetables and

Medicinal Plants, Directorate General of Horticulture, Ministry of Agriculture explained, currently the demand for mushrooms in Indonesia is very high, however, the annual production has only reached 33 tons (Kementerian Pertanian RI, 2022). This amount is certainly not sufficient to meet all domestic and foreign demands. Moreover, the production of mushroom plants on the island of Sumatra for the past 2 years has decreased, namely in 2020 and 2021. This of course has a huge impact on the sales sector because the demand for goods is high while production is low, so it will cause demand to be unfulfilled (Hastuti et al., 2020). In this case, it is necessary to predict mushroom production in Sumatra in the future so that the government on the island of Sumatra has clear data references to determine policies, and make appropriate steps so that the production of mushroom plants on the island of Sumatra does not continue to decline, and is even able to increase so that it will be able to move the wheels of the economy on the island of Sumatra (Retiwiranti, 2018). The Conjugate Gradient Algorithm is a technique that will be used in this study (Sari, 2017).

There are many techniques in computer science that can deal with complex problems, as evidenced by the many analyzes that have been made. Examples include the field of decision support systems (Ningsih et al., 2019; Sari et al., 2019; Sundari et al., 2019), the field of data mining (Damanik et al., 2019; Elisa, 2017; Sudirman et al., 2018) , as well as research in the field of artificial neural networks (Manurung et al., 2022; Marlina & Arifin, 2021; Sinaga et al., 2019). Knowing the performance of an artificial neural network algorithm is very important to make it easier to determine the prediction model to be used and produce accurate forecasting data, several studies have been carried out, such as the study of Wanto et al. (2017) used backpropagation and the Fletcher-Reeves algorithm to solve the problem of forecasting the consumer price index (Wanto et al., 2017). There is also research from Tinambunan et al. (2020) which uses the Polak-Ribiere algorithm to improve the performance of the standard backpropagation algorithm in solving population problems. In this study, the Fletceher-Reeves algorithm minimizes iteration and usage time (Tinambunan et al., 2020).

Based on related descriptions from previous studies, this paper will predict the production of mushroom plants in Sumatra using the conjugate gradient algorithm. This dataset is only used to aid in the verification and process of measuring the performance of the algorithm. The purpose of this study was to optimize the accuracy and performance of the algorithm in finding the best results to solve the problem of predicting mushroom production in Sumatra.

2. RESEARCH METHOD

2.1 Artificial Neural Network

Artificial Neural Network (ANN) is a system created using the concept of representing the workings of the human brain so that it can function like the learning process in the human brain. (Agus Perdana Windarto, 2017). Because this neural network is implemented with a computer program and can complete all calculation processes, this system is called artificial or artificial. (Mira Febrina, Faula Arina, 2022). Artificial Neural Networks (ANN) has a fundamental trait, namely intelligence. Analogous to human intelligence ANN can solve difficult problems if solved using conventional computing (Edisar, 2015). ANN is designed to resemble the ability in humans who can fully understand all pattern information and can adapt processing patterns well (Batubara & Awangga, 2020).

2.2 Backpropagation

Backpropagation is part of ANN which has the ability to be able to recognize patterns during training and can also provide the right action for the right input pattern with the pattern used during training (Revi et al., 2018). The three phases of backpropagation training include a feed forward phase, a back propagation phase, and a weight modification phase. The input is calculated forward starting from the input layer to the output layer. In the backpropagation stage, each result unit gets an objective example related to the sample information to resolve the error value. Errors are propagated in the opposite direction. For the weight modification phase, it is used to reduce errors that occur (Nurmila et al., 2010). The three phases are repeated continuously until the termination condition is met (Nurmila et al., 2010).

2.3 Data collection

Quantitative techniques are used in the data collection process in this paper, namely data on Vegetable Crop Production in Indonesia from 2014-2021 (Aan Melinda, 2021). However, this study only took data on mushroom production on the island of Sumatra, namely Aceh, North Sumatra, West Sumatra, Riau, Jambi, South Sumatra, Bengkulu, Lampung, Kep. Bangka Belitung, and Kep. Riau. The data comes from the BPS (Indonesian Central Statistics Agency) website.

Table 1. Mushroom Plant Production Data on Sumatra Island (Kg)

Region	2014	2015	2016	2017	2018	2019	2020	2021
ACEH	8,835.00	12,872.00	4,599.00	338.00	49,219.00	26,738.00	2,674.00	82.00
NORTH SUMATRA	15,954.00	16,310.00	18,477.00	1,955.00	23,493.00	37,119.00	3,712.00	304.00
WEST SUMATRA	139.998.00	116,513.00	58,727.00	3,246.00	40,110.00	98,066.00	9,807.00	203.00
RIAU	260,102.00	739,234.00	95,571.00	19,813.0(	72,849.00	49,952.00	4,995.00	847.00
JAMBI	22,923.00	15,318.00	15,648.00	1.550.00	17,417.00	16,514.00	1,651.00	97.00
SOUTH SUMATRA	418,323.00	231.871.00	25,101.00	5,138.00	91,776.00	205.732.00	20,573.0(	703.00
BENGKULU	4,909.00	3,812.00	5,039.00	1,734.00	8,5000.00	14.003.00	1,400.00	2,748.0
LAMPUNG	300,550.00	367,092.00	231.854.00	13,382.0(	280,971.00	175,623.00	17,562.0(	4,817.0
KEEP. BANGKA BELITUNG	56.00	24.00	90.00	8.00	672.00	2,119.00	212.00	33.00
KEEP. RIAU	171.00	527.00	598.00	61.00	664.00	2,052.00	205.00	147.00

2.4 Research Stages

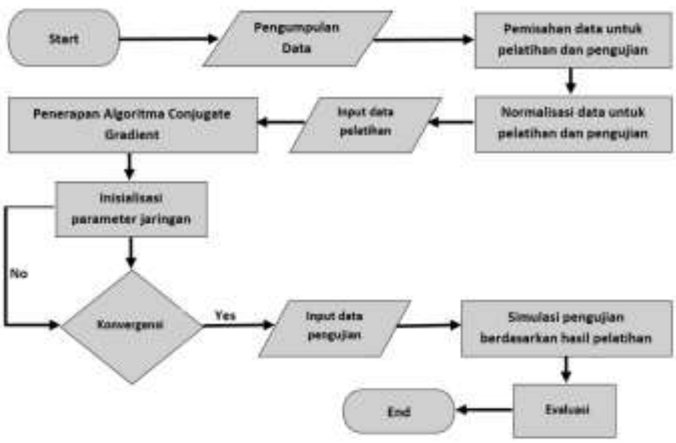

Figure 1. Research Stages

From Figure 1, the first step is to collect research data. Next is to divide the research data into 2 parts, namely training and testing data. Data for 2014-2016 with a target in 2017 which will be used as training data, and for data for 2018-2020 with a target in 2021 which will be used as test data.

After being divided into 2 parts, the data will be normalized using the sigmoid ability or function. The next step is to use the equation (1) to normalize the training and testing data (Manurung et al., 2022), (Sinaga et al., 2019), (Sugiantari & Budiantara, 2013), (Siregar et al., 2019) .

$$x' = \frac{0.8(x-a)}{b-a} + 0.1 \tag{1}$$

Which: X' is the result of normalized data, values 0.8 and 0.1 are values that have been set for the normalization formula (Default Value), X is the data to be normalized, b is the Max value (highest) in the initial data and a is the Min (lowest) value in the initial data. The next step is to input the training data that has been normalized previously in the Matlab 2011b software for processing. Then proceed to the process of applying the Conjugate Gradient Algorithm. Then proceed with creating a multi-layer neural network that uses the tansig and logsig functions. Then use the training function (trainscg) to initialize the network parameters. Then enter the command to run the training process and see the resulting performance. If the training results reach convergence, it is continued by entering test data (test data), if not, it will return to the network parameter initialization phase. The next step is to simulate the test data based on the results of the training. The last stage is to find the best architectural model based on the lowest Performance/MSE testing results.

3 RESULTS AND DISCUSSIONS

3.1 Normalized Data Results

Table 2 below shows the results of the normalization of the training data used, namely 2014 to 2016, and 2017 as targets. The data is quoted from the data in table 1. Then normalization will be carried out on the data using the equation (1).

Table 2. Training Data Normalization Results

NO	2014	2015	2016	2017(target)
1	0.1096	0.1139	0.1050	0.1004
2	0.1173	0.1176	0.1200	0.1021
3	0.2515	0.2261	0.1635	0.1035
4	0.3815	0.9000	0.2034	0.1214
5	0.1248	0.1166	0.1169	0.1017
6	0.5527	0.3509	0.1272	0.1056
7	0.1053	0.1041	0.1054	0.1019
8	0.4253	0.4973	0.3509	0.1145
9	0.1001	0.1000	0.1001	0.1000
10	0.1002	0.1006	0.1006	0.1001

Table 3 below shows the results of the normalization of the test data used, namely 2018 to 2020, and 2021 as targets. The data is quoted from the data in table 1. Then normalization will be carried out on the data using the equation (1).

Table 3. Testing Data Normalization Results

NO	2018	2019	2020	2021(target)
1	0.2401	0.1760	0.1075	0.1001
2	0.1668	0.2056	0.1105	0.1008
3	0.2141	0.3792	0.1278	0.1005
4	0.3074	0.2421	0.1141	0.1023
5	0.1495	0.1469	0.1046	0.1002
6	0.3612	0.6857	0.1585	0.1019
7	0.1241	0.1398	0.1039	0.1077
8	0.9000	0.6000	0.1499	0.1136
9	0.1018	0.1059	0.1005	0.1000
10	0.1018	0.1057	0.1005	0.1003

3.2 Training and Testing

Data processing is carried out using the matlab 2011b application which aims to determine the best architectural model. The method used in the architecture is the Conjugate Gradient method. There are 5 architectures used in this paper, namely: 3-10-1, 3-15-1, 3-20-1, 3-25-1 and 3-30-1. The first data structure in the model is called Input (3), the second data is called Hidden (10, 15, 20, 25, 30) and the third data is called Output (1). The parameters of the Conjugate Gradient algorithm used are shown in Figure 2.

```
% Nilai parameter default Scaled conjugate gradient backpropagation (trainscg)
net.trainParam.epochs = 1000;
net.trainParam.show = 25;
net.trainParam.showCommandLine = 0;
net.trainParam.showWindow = 1;
net.trainParam.goal = 0;
net.trainParam.time = inf;
net.trainParam.min_grad = 1e-6;
net.trainParam.max_fail = 5;
net.trainParam.sigma = 5.0e-5;
net.trainParam.lambda = 5.0e-7;
```

Figure 2. Conjugate Gradient Parameters

3.2.1 Model Training and Testing 3-10-1

The results of the 3-10-1 architectural model can be seen with the epoch results of 132 iterations. The results of the training and testing are shown in tables 4 and 5.

Table 4. Training Results

No	X1	X2	X3	Target(Y1)	Epoch 132		
					actual	Error	perfect
1	0.1096	0.1139	0.1050	0.1004	0.1002	0.0002	0.00000031
2	0.1173	0.1176	0.1200	0.1021	0.1028	-0.0007	
3	0.2515	0.2261	0.1635	0.1035	0.1035	0.0000	
4	0.3815	0.9000	0.2034	0.1214	0.1214	0.0000	
5	0.1248	0.1166	0.1169	0.1017	0.1007	0.0010	
6	0.5527	0.3509	0.1272	0.1056	0.1056	0.0000	
7	0.1053	0.1041	0.1054	0.1019	0.1011	0.0008	
8	0.4253	0.4973	0.3509	0.1145	0.1145	0.0000	
9	0.1001	0.1000	0.1001	0.1000	0.1007	-0.0007	
10	0.1002	0.1006	0.1006	0.1001	0.1008	-0.0007	

Table 5. Test result

No	X5	X6	X7	Target(Y2)	epoch		
					actual	Error	perfect
1	0.2401	0.1760	0.1075	0.1001	0.8005	0.0196	0.00055034
2	0.1668	0.2056	0.1105	0.1008	0.0938	0.0070	
3	0.2141	0.3792	0.1278	0.1005	0.0819	0.0186	
4	0.3074	0.2421	0.1141	0.1023	0.0748	0.0275	
5	0.1495	0.1469	0.1046	0.1002	0.0949	0.0053	
6	0.3612	0.6857	0.1585	0.1019	0.1629	-0.0610	
7	0.1241	0.1398	0.1039	0.1077	0.0980	0.0097	
8	0.9000	0.6000	0.1499	0.1136	0.1025	0.0111	
9	0.1018	0.1059	0.1005	0.1000	0.1004	-0.0004	
10	0.1018	0.1057	0.1005	0.1003	0.1004	-0.0001	

3.2.2 Model Training and Testing 3-15-1

The results of the 3-15-1 architectural model can be seen with the epoch results of 128 iterations. The results of the training and testing are shown in Tables 6 and 7.

Table 6. Training Results

No	X1	X2	X3	Target(Y1)	Epoch 128 actual	Error	perfect
1	0.1096	0.1139	0.1050	0.1004	0.0994	0.0010	0.00000018
2	0.1173	0.1176	0.1200	0.1021	0.1019	0.0002	
3	0.2515	0.2261	0.1635	0.1035	0.1035	0.0000	
4	0.3815	0.9000	0.2034	0.1214	0.1214	0.0000	
5	0.1248	0.1166	0.1169	0.1017	0.1022	-0.0005	
6	0.5527	0.3509	0.1272	0.1056	0.1056	0.0000	
7	0.1053	0.1041	0.1054	0.1019	0.1016	0.0003	
8	0.4253	0.4973	0.3509	0.1145	0.1145	0.0000	
9	0.1001	0.1000	0.1001	0.1000	0.1005	-0.0005	
10	0.1002	0.1006	0.1006	0.1001	0.1005	-0.0004	

Table 7. Test result

No	X5	X6	X7	Target(Y2)	epoch actual	Error	perfect
1	0.2401	0.1760	0.1075	0.1001	0.1657	-0.0656	0.00578390
2	0.1668	0.2056	0.1105	0.1008	0.1052	-0.0044	
3	0.2141	0.3792	0.1278	0.1005	0.2447	-0.1442	
4	0.3074	0.2421	0.1141	0.1023	0.2349	-0.1326	
5	0.1495	0.1469	0.1046	0.1002	0.1001	0.0001	
6	0.3612	0.6857	0.1585	0.1019	0.2050	-0.1031	
7	0.1241	0.1398	0.1039	0.1077	0.0950	0.0127	
8	0.9000	0.6000	0.1499	0.1136	0.1796	-0.0660	
9	0.1018	0.1059	0.1005	0.1000	0.0992	0.0008	
10	0.1018	0.1057	0.1005	0.1003	0.0992	0.0011	

3.2.3 Model Training and Testing 3-20-1

The results of the 3-20-1 architectural model can be seen with the epoch results of 67 iterations. The results of the training and testing are shown in tables 8 and 9.

Table 8. Training Results

No	X1	X2	X3	Target(Y1)	Epoch 67 actual	Error	perfect
1	0.1096	0.1139	0.1050	0.1004	0.1006	-0.0002	0.00000028
2	0.1173	0.1176	0.1200	0.1021	0.1013	0.0008	
3	0.2515	0.2261	0.1635	0.1035	0.1035	0.0000	
4	0.3815	0.9000	0.2034	0.1214	0.1214	0.0000	
5	0.1248	0.1166	0.1169	0.1017	0.1024	-0.0007	
6	0.5527	0.3509	0.1272	0.1056	0.1056	0.0000	
7	0.1053	0.1041	0.1054	0.1019	0.1008	0.0011	
8	0.4253	0.4973	0.3509	0.1145	0.1145	0.0000	
9	0.1001	0.1000	0.1001	0.1000	0.1005	-0.0005	
10	0.1002	0.1006	0.1006	0.1001	0.1005	-0.0004	

Table 9. Test result

No	X5	X6	X7	Target(Y2)	epoch actual	Error	perfect
1	0.2401	0.1760	0.1075	0.1001	0.1291	-0.0290	0.05681989
2	0.1668	0.2056	0.1105	0.1008	0.0982	0.0026	
3	0.2141	0.3792	0.1278	0.1005	0.0756	0.0249	
4	0.3074	0.2421	0.1141	0.1023	0.1427	-0.0404	
5	0.1495	0.1469	0.1046	0.1002	0.1039	-0.0037	
6	0.3612	0.6857	0.1585	0.1019	0.2600	-0.1581	
7	0.1241	0.1398	0.1039	0.1077	0.1005	0.0072	
8	0.9000	0.6000	0.1499	0.1136	0.8485	-0.7349	
9	0.1018	0.1059	0.1005	0.1000	0.1003	-0.0003	
10	0.1018	0.1057	0.1005	0.1003	0.1003	0.0000	

3.2.4 Model 3-25-1 . Training and Testing

The results of the 3-20-1 architectural model can be seen with the epoch results of 230 iterations. The results of the training and testing are shown in tables 10 and 11.

Table 10. Training Results

No	X1	X2	X3	Target(Y1)	Epoch 230		
					actual	Error	perfect
1	0.1096	0.1139	0.1050	0.1004	0.1004	0.0000	0.00000026
2	0.1173	0.1176	0.1200	0.1021	0.1024	-0.0003	
3	0.2515	0.2261	0.1635	0.1035	0.1035	0.0000	
4	0.3815	0.9000	0.2034	0.1214	0.1214	0.0000	
5	0.1248	0.1166	0.1169	0.1017	0.1016	0.0001	
6	0.5527	0.3509	0.1272	0.1056	0.1056	0.0000	
7	0.1053	0.1041	0.1054	0.1019	0.1005	0.0014	
8	0.4253	0.4973	0.3509	0.1145	0.1145	0.0000	
9	0.1001	0.1000	0.1001	0.1000	0.1006	-0.0006	
10	0.1002	0.1006	0.1006	0.1001	0.1006	-0.0005	

Table 11. Test result

No	X5	X6	X7	Target(Y2)	epoch		
					actual	Error	perfect
1	0.2401	0.1760	0.1075	0.1001	0.1498	-0.0497	0.00643667
2	0.1668	0.2056	0.1105	0.1008	0.1242	-0.0234	
3	0.2141	0.3792	0.1278	0.1005	0.2983	-0.1978	
4	0.3074	0.2421	0.1141	0.1023	0.2317	-0.1294	
5	0.1495	0.1469	0.1046	0.1002	0.1084	-0.0082	
6	0.3612	0.6857	0.1585	0.1019	0.1712	-0.0693	
7	0.1241	0.1398	0.1039	0.1077	0.1030	0.0047	
8	0.9000	0.6000	0.1499	0.1136	0.1379	-0.0243	
9	0.1018	0.1059	0.1005	0.1000	0.1004	-0.0004	
10	0.1018	0.1057	0.1005	0.1003	0.1004	-0.0001	

3.2.5 Model 3-30-1 . Training and Testing

The results of the 3-20-1 architectural model can be seen with the epoch results of 104 iterations. The results of training and testing are shown in tables 12 and 13.

Table 12. Training Results

No	X1	X2	X3	Target(Y1)	Epoch 104		
					actual	Error	perfect
1	0.1096	0.1139	0.1050	0.1004	0.1004	0.0000	0.00000025
2	0.1173	0.1176	0.1200	0.1021	0.1024	-0.0003	
3	0.2515	0.2261	0.1635	0.1035	0.1035	0.0000	
4	0.3815	0.9000	0.2034	0.1214	0.1214	0.0000	
5	0.1248	0.1166	0.1169	0.1017	0.1018	-0.0001	
6	0.5527	0.3509	0.1272	0.1056	0.1056	0.0000	
7	0.1053	0.1041	0.1054	0.1019	0.1005	0.0014	
8	0.4253	0.4973	0.3509	0.1145	0.1145	0.0000	
9	0.1001	0.1000	0.1001	0.1000	0.1006	-0.0006	
10	0.1002	0.1006	0.1006	0.1001	0.1005	-0.0004	

Table 13. Test result

No	X5	X6	X7	Target(Y2)	epoch		
					actual	Error	perfect
1	0.2401	0.1760	0.1075	0.1001	0.3444	-0.2443	0.02780333
2	0.1668	0.2056	0.1105	0.1008	0.1267	-0.0259	
3	0.2141	0.3792	0.1278	0.1005	0.1674	-0.0669	
4	0.3074	0.2421	0.1141	0.1023	0.5347	-0.4324	

No	X5	X6	X7	Target(Y2)		epoch	
					actual	Error	perfect
5	0.1495	0.1469	0.1046	0.1002	0.1153	-0.0151	
6	0.3612	0.6857	0.1585	0.1019	0.2595	-0.1576	
7	0.1241	0.1398	0.1039	0.1077	0.1036	0.0041	
8	0.9000	0.6000	0.1499	0.1136	0.0792	0.0344	
9	0.1018	0.1059	0.1005	0.1000	0.1003	-0.0003	
10	0.1018	0.1057	0.1005	0.1003	0.1003	0.0000	

3.3 Evaluation

Based on the five models that have been carried out in the training and testing process, namely 3-10-1, 3-15-1, 3-20-1, 3-25-1 and 3-30-1 using Matlab 2011b and Microsoft Excel applications, then the most optimal architectural model is 3-10-1 with a Performance/MSE test value of 0.00055034. This value is the smallest of the 5 architectural models after the training and testing process.

Table 14. Comparison of Overall Model Results

Algorithm	Architecture	Training Function	Epoch (Iteration)	MSE Training	MSE Testing/Performance
Conjugate Gradient	3-10-1	trainscg	132	0.00000031	0.00055034
	3-15-1	trainscg	128	0.00000018	0.00578390
	3-20-1	trainscg	67	0.00000028	0.05681989
	3-25-1	trainscg	230	0.00000026	0.00643667
	3-30-1	trainscg	104	0.00000025	0.02780333

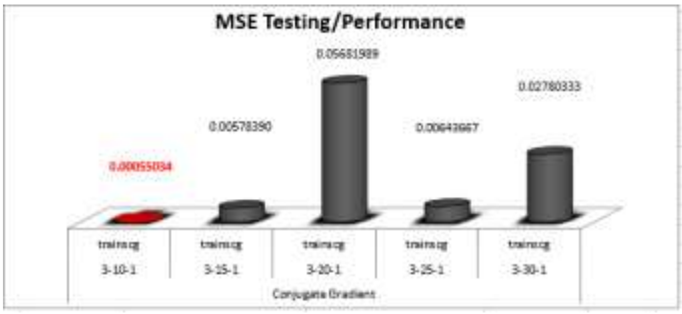

Figure 3.MSE Testing / Performance Comparison Chart

4. Conclusion

From the results and discussions that have been described in this paper, the conclusion that can be drawn is that the Conjugate Gradient Algorithm is capable and can be used to predict Mushroom Plant Production in Sumatra, as one step in helping the government so that the government on the island of Sumatra has clear data references to determine policies and make the right steps, so that the production of mushroom plants on the island of Sumatra does not continue to decline. In this study, the Conjugate Gradient Algorithm was able to produce the best architectural model, namely 3-10-1, where the Performance/MSE test value was 0.00055034.

ACKNOWLEDGEMENTS

We would like to thank all those who have contributed to this research, so that the research can be carried out properly.

REFERENCES

AAN MELINDA, S. (2021). *PERSEPSI IBU PKK DI DESA PANGKUL KECAMATAN CAMBAI KOTA PRABUMULIH TENTANG PESAN DAKWAH DALAM SINETRON JODOH WASIAT BAPAK DI ANTV*. UIN RADEN FATAH PALEMBANG.

Batubara, N. A., & Awangga, R. M. (2020). *Tutorial Object Detection Plate Number With Convolution Neural Network (CNN)* (Vol. 1). Kreatif.

Edisar, M. (2015). *Aplikasi Jaringan Syaraf Tiruan (Jst) Untuk Estimasi Kecepatan Gelombang Kompressi (P) Pada Lapisan Batuan Bawah Permukaan*.

Hastuti, P., Harefa, D. N., & Napitupulu, J. I. M. (2020). Tinjauan kebijakan pemberlakuan lockdown, phk, psbb sebagai antisipasi penyebaran covid-19 terhadap stabilitas sistem moneter. *Prosiding WEBINAR Fakultas Ekonomi Unimed "Strategi Dunia Usaha Menyikapi Status Indonesia Sebagai Negara Maju: Pra Dan Pasca Covid-19,"* 57–70.

https://www.google.com/books?hl=id&lr=&id=NLl_EAAAQBAJ&oi=fnd&pg=PA1&dq=Kementerian+Pertanian+RI.+(2022).+Prospek+Bisnis+Jamur+di+tengah+Pandemi+Covid+-+19.+Www.Pertanian.Go.Id.&ots=xLZdtL2ybU&sig=vhd2pMDfjQtQHMXkv2sWxgxtfxw

Lianah, M. P. (2020). *Budidaya Jamur Pangan Konsumsi Lokal*. Alinea Media Dipantara.

Nurmila, N., Sugiharto, A., & Sarwoko, E. A. (2010). Algoritma Back Propagation Neural Network Untuk Pengenalan Pola Karakter Huruf Jawa. *Jurnal Masyarakat Informatika*, *1*(1), 1–10. https://ejournal.undip.ac.id/index.php/jmasif/article/download/74/521

Pujiati, W. (2018). *Identifikasi Jamur Aspergillus sp Pada Tepung Terigu Yang Dijual Secara Terbuka (Studi di Pasar Legi Jombang)*. STIKES Insan Cendekia Medika Jombang.

Retiwiranti, M. (2018). *Analisis Peran Ekonomi Kreatif Pada Masyarakat Dalam Meningkatkan Pendapatan Rumah Tangga Melalui Budidaya Tanaman Biofarmaka Dalam Perspektif Ekonomi Islam (Studi Pada Petani Kencur Desa Fajar Asri Kecamatan Seputih Agung Kabupaten Lampung Tengah)*. UIN Raden Intan Lampung.

Revi, A., Solikhun, S., & Parlina, I. (2018). Jaringan Syaraf Tiruan Dalam Memprediksi Tingkat Pertumbuhan Industri Mikro Dan Kecil Berdasarkan Provinsi. *Teknika*, *7*(2), 129–137. http://ejournal.ikado.ac.id/index.php/teknika/article/view/123

Sari, Y. (2017). Prediksi harga emas menggunakan metode neural network backpropagation algoritma conjugate gradient. *Jurnal Eltikom*, *1*(2).

Savitri, A. (2016). *Tanaman Ajaib! Basi Penyakit dengan TOGA (Tanaman Obat Keluarga)*. Bibit Publisher.

Tinambunan, M. H., Nababan, E. B., & Nasution, B. B. (2020). Conjugate Gradient Polak Ribiere in Improving Performance in Predicting Population Backpropagation. *IOP Conference Series: Materials Science and Engineering*, *835*(1). https://iopscience.iop.org/article/10.1088/1757-899X/509/1/012119/meta

Wanto, A., Zarlis, M., Sawaluddin, & Hartama, D. (2017). Analysis of Artificial Neural Network Backpropagation Using Conjugate Gradient Fletcher Reeves in the Predicting Process. *Journal of Physics: Conference Series*, *930*(1).https://iopscience.iop.org/article/10.1088/1742-6596/930/1/012018/meta

Waris, W., & Khasanah, N. (2019). Optimalisasi Potensi Ekonomi Masyarakat Melalui Budidaya Jamur Tiram. *Dimas: Jurnal Pemikiran Agama Untuk Pemberdayaan*, *18*(2), 159–176.