Best Model and Performance of Bayesian Regularization Method for Data Prediction

Pratomo Setiaji1, Bambang Widjanarko2, Yuda Syahidin3, Hidayatulah Himawan4, Nurintan Asyiah Siregar5, Harmayani6, Lukman Samboteng7, Nur’Ainun Gulo8, and Reni Kartikaningsih9

1Universitas Muria Kudus, Kudus, Indonesia. Email: pratomo.setiaji@umk.ac.id
2Universitas Pembangunan Panca Budi, Medan, Indonesia. Email: bambangwidjanarko@dosen.pancabudi.ac.id
3Politeknik Piksi Ganesha, Indonesia. Email: yudasy@gmail.com
4UPN Veteran Yogyakarta, Indonesia. Email: if.iwan@upnyk.ac.id
5Universitas Labuhanbatu, Indonesia. Email: nurintanasyahsiregar@gmail.com
6Universitas Asahan, Kisaran, Indonesia
7STIA LAN Makassar, Indonesia. Email: lukmansamboteng@yahoo.co.id
8Universitas Labuhanbatu, Indonesia. Email: nurainungulo7@gmail.com
9Universitas Labuhanbatu, Indonesia. Email: cahyaningrembulan0@gmail.com

Abstract. The backpropagation algorithm has many training and activation functions that can be used to influence or maximize prediction results, all of which have their respective advantages and disadvantages. The purpose of this paper is to analyze one of the training functions of the backpropagation algorithm which can be used as a reference for use in data prediction problems in the form of models and best performance. The training function is the Bayesian Regularization method. This method is able to train the network by optimizing the Levenberg-Marquardt by updating the bias and weights. The research dataset used to analyze the data in this paper is Formal Education Participation in Indonesia 2015-2020 which consists of the School Participation Rate, the Gross Enrollment Rate, and the Pure Enrollment Rate. The 2015-2016 dataset is used as training data with a 2017 target, while the 2018-2019 dataset is the test data with a 2020 target. The models used are 2-10-1, 2-15-1, and 2-20-1. Based on the analysis and calculation process, the results of the 2-15-1 model are the best with an epoch of 217 iterations and an MSE of 0.00002945, this is because the epoch is not too large and has the smallest MSE compared to the other 2 models.

1. Introducing

Artificial Neural Networks (ANN) have characteristics that are almost similar to Biological Neural Networks which are determined by 3 things: network architecture, the method of determining the connecting weight (training / learning method) and the activation function [1]. There are many types of ANN algorithms, including perceptron, backpropagation, Learning vector quantization (LVQ), probabilistic neural network, Hopfield, radial base network [2]–[7], and others, all of which have their own characteristics. The ANN algorithm discussed in this paper is Bayesian Regularization.
Backpropagation or simply called Bayesian Regularization, which is an ANN method that is able to work systematically by training multiplayer networks using mathematical science based on the network architecture models developed. The training function of the Bayesian Regularization method is able to train the network by optimizing the Levenberg-Marquardt by updating the bias and weights [8]. Data that has been trained properly will provide appropriate output if it is given input that is different from the architectural pattern used in training. This generalization property makes training more efficient because it does not require a very long time like conventional backpropagation algorithms.

This paper will discuss and analyze the training function of the ANN algorithm using the Bayesian Regularization method to solve prediction problems, in order to obtain the best model and performance that can be used as a reference for obtaining predictive results. Currently, prediction methods that use computational, statistical, and experiential data are very interesting to research, especially when using soft computing and artificial intelligence tools such as artificial neural networks (ANN) which are well known [9], one of them is the Bayesian Regularization method. Bayesian Regularization technique makes it possible to reduce overfitting problems that may arise from the selection of a network model [10] and able to train the network by optimizing the Levenberg-Marquardt by updating the bias and weights. Bayesian rule regularization process is usually carried out through previously informative distributions and limits parameter freedom [11].

Bayesian regularization method has been used widely to solve many complex problems. M Byrd, et al (2020) discusses the Bayesian regularization of the Gaussian graphical model with measurement errors and proposes a correction methodology for the Gaussian graphical model when contaminated with additive measurement errors. The core solution to this problem involves the use of an imputation-regularization algorithm to generate the true value of the underlying process with consistent precision matrix estimates [12]. L Gan, et al (2020) in his paper using bayesian regularization for graphical models with unequal shrinkage. Based on the context of the graphical model, the results of this study indicate that the spike-and-slab prior with the Laplace distribution provides an adaptive penalty leading to better theoretical and empirical performance compared to state-of-the-art methods [13]. D Hu, et al (2020) in their paper proposed a forecasting method to develop a predictive model for the quantity parameters of electric vehicles in each period. The prediction tool used is a feed-forward neural network based on Bayesian learning regularization backpropagation as an input data model so that planning is more reliable and accurate. Due to long charging times, electric vehicles tend to queue long. So that the congestion level limitation is imposed for each station to adjust its capacity. In addition, two Greedy Algorithms are proposed, which are proven by the complementary test results. Therefore, both algorithms are adopted simultaneously to solve the problem. The end result is the application of this method in the planning of filling stations in Cixi city [14]. I Khan, et al (2020) discuss a new computational paradigm by leveraging the power of feed-forward neural networks (ANN) with the Levenberg-Marquardt Method (LMM), and the Bayesian Regularization Method (BRM) based on IVBs based backpropagation from the pantograph delay differential equation. linear / nonlinear (LP / NP-DDEs). The dataset for training, testing, and validation is created with reference to known standard LP / NP-DDEs solutions. ANN is implemented using this dataset to estimate system modeling on merit function based on mean squared errors, while adjustable parameter learning is carried out with the efficacy of LMM (ANN-LMM) and BRM (ANN-BRM). The performance of the algorithms designed by ANN-LMM and ANN-BRM on first, second, and third-order NP-FDE IVPs are verified by reaching a good agreement with available solutions that have accuracy in the range 10−5 to 10−8 and are further supported via error histograms and regression measures [15]. E Sariev and G Germano (2020) propose an improved Bayesian regularization approach to train ANN and compare it to classical regularization which relies on backpropagation algorithms to train feedforward networks. They investigated different network architectures and tested the classification accuracy of three data sets. As a result Profitability, leverage and liquidity emerge as important categories of drivers of financial default [16]. Based on these previous studies, this paper proposes a training function using the Bayesian Regularization method combined with the binary sigmoid (logsig) activation function and the linear (purelin) function to predict data. The method is analyzed by training
and testing on times-series data on formal education participation in Indonesia, to find the best model and performance.

2. Methodology

2.1. Experiment Dataset

The experimental dataset used in this paper is data on participation in formal education in Indonesia obtained from the Indonesian Central Bureau of Statistics. This data is quantitative data with variable the School Participation Rate (SPR) aged 7-12 years, School Participation Rate (SPR) aged 13-15 years, School Participation Rate (SPR) aged 16-18 years, School Participation Rate (SPR) aged 19-24 years, Gross Enrollment Rate (GER) SD / MI, Gross Enrollment Rate (GER) SMP / MTs, Gross Enrollment Rate (GER) SM / MA, Gross Enrollment Rate (GER) PT aged 19-24 years, Net Participation Rate (NPR) SD/MI, Net Participation Rate (NPR) SMP / MTs, Net Participation Rate (NPR) SM / MA and Net Participation Rate (NPR) PT aged 19-24 years.

| No  | Formal Educational Participation | 2015   | 2016   | 2017   | 2018   | 2019   | 2020   |
|-----|----------------------------------|--------|--------|--------|--------|--------|--------|
| 1   | School Participation Rate (SPR) 7-12 th | 98,57  | 98,98  | 99,08  | 99,11  | 99,17  | 99,21  |
| 2   | School Participation Rate (SPR) 13-15 th | 94,25  | 94,79  | 94,98  | 95,23  | 95,43  | 95,52  |
| 3   | School Participation Rate (SPR) 16-18 th | 70,26  | 70,68  | 71,20  | 71,82  | 71,92  | 71,44  |
| 4   | School Participation Rate (SPR) 19-24 th | 22,77  | 23,80  | 24,67  | 24,29  | 23,28  | 22,53  |
| 5   | Gross Enrollment Rate (GER) SD/MI | 109,94 | 109,20 | 108,43 | 108,48 | 107,36 | 105,97 |
| 6   | Gross Enrollment Rate (GER) SMP/MTs | 90,63  | 89,98  | 90,00  | 91,23  | 90,20  | 88,94  |
| 7   | Gross Enrollment Rate (GER) SM/MA | 77,39  | 80,44  | 82,25  | 80,11  | 79,94  | 78,61  |
| 8   | Gross Enrollment Rate (GER) PT 19-24 th | 20,89  | 23,44  | 25,00  | 25,12  | 25,13  | 25,50  |
| 9   | Net Participation Rate (NPR) SD/MI | 96,20  | 96,71  | 97,10  | 97,48  | 97,58  | 97,65  |
| 10  | Net Participation Rate (NPR) SMP/MTs | 77,45  | 77,89  | 78,30  | 78,75  | 79,35  | 80,02  |
| 11  | Net Participation Rate (NPR) SM/MA | 59,46  | 59,85  | 60,19  | 60,53  | 60,70  | 61,03  |
| 12  | Net Participation Rate (NPR) PT 19-24 th | 17,34  | 17,91  | 18,62  | 18,59  | 18,85  | 19,32  |

Source : Central Bureau of Statistics

2.2. Research Stages

Stages of research carried out to forecast the level of inflation growth in Indonesia based on expenditure groups include:

a. Collect the Research dataset to be used.
b. Preprocessing. The knowledge is then normalized using the following equation [17]–[21]:

\[ x' = \frac{a \times (x - a) + 0.1}{b - a} \]

Formula description: \( x' \): is the normalization result, \( x \): is the normalized info, \( a \): is the lowest value, and \( b \): is the highest value.

Then the data is split into two sections, namely training and testing.
c. Determine the concept of the network architecture to be used for the training and testing process.
d. Analyze the architectural model used.
e. Choose the best architectural model and performance.

3. Results and Discussion

3.1. Normalizing data

The formal education participation dataset presented in table 1 is normalized first using the formula in Equation (1).

| Formal Educational Participation | Month |
|---------------------------------|-------|
|                                 |       |
After the data has been normalized, the next step is to divide the data into 2 groups. Group one as training data and the second group as test data. For training data using data for 2015 (X1) - 2016 (X2) with a target of 2017 (Y). While the test data is taken from 2018 (X1) - 2019 (X2) with a target of 2020 (Y). To help data analysis, Matlab 2011b and Microsoft Excel tools were used with the model to be analyzed by Bayesian regularization 2-10-1, 2-15-1, and 2-20-1. Broadly speaking, the parameters in this paper use the binary sigmoid activation function (logsig) and the linear function (purelin) with an epoch = 1000 limit, while the supporting parameters are in accordance with the default parameters of the Bayesian regularization technique in Matlab.

3.2. Architectural Model
The following will present the results of the analysis and calculations from the comparison of the 2-10-1, 2-15-1 and 2-20-1 models used.

a. Model 2-10-1

|     | 2015       | 2016       | 2017       | 2018       | 2019       | 2020       |
|-----|------------|------------|------------|------------|------------|------------|
| 1   | 0.80177    | 0.80531    | 0.80618    | 0.81661    | 0.81714    | 0.81750    |
| 2   | 0.76445    | 0.76911    | 0.77076    | 0.78208    | 0.78386    | 0.78466    |
| 3   | 0.55719    | 0.56082    | 0.56531    | 0.57373    | 0.57462    | 0.57035    |
| 4   | 0.14691    | 0.15581    | 0.16333    | 0.15073    | 0.14174    | 0.13507    |
| 5   | 0.90000    | 0.89361    | 0.88695    | 0.90000    | 0.89003    | 0.87766    |
| 6   | 0.73317    | 0.72756    | 0.72773    | 0.74648    | 0.73731    | 0.72610    |
| 7   | 0.61879    | 0.64514    | 0.66078    | 0.64751    | 0.64600    | 0.63416    |
| 8   | 0.13067    | 0.15270    | 0.16618    | 0.15812    | 0.15820    | 0.16150    |
| 9   | 0.78130    | 0.78570    | 0.78907    | 0.80210    | 0.80299    | 0.80362    |
| 10  | 0.61931    | 0.62311    | 0.62665    | 0.63541    | 0.64075    | 0.64671    |
| 11  | 0.46389    | 0.46726    | 0.47019    | 0.47326    | 0.47477    | 0.47771    |
| 12  | 0.10000    | 0.10492    | 0.11106    | 0.10000    | 0.10231    | 0.10650    |

Figure 1. Results of Training Model 2-10-1

Figure 1 is the result of training with Matlab for model 2-10-1. The training of the model resulted in an Epoch of 263 iterations.
### Table 3. Training Model 2-10-1

| No | X1     | X2     | Target (Y) | Actual | Error   | SSE   | Performance |
|----|--------|--------|------------|--------|---------|-------|-------------|
| 1  | 0.80177| 0.80531| 0.80618    | 0.80670| -0.00052| 0.00000027|
| 2  | 0.76445| 0.76911| 0.77076    | 0.77210| -0.00134| 0.00000181|
| 3  | 0.55719| 0.56082| 0.5631     | 0.56500| 0.00031  | 0.00000010|
| 4  | 0.14691| 0.15581| 0.1633     | 0.16280| 0.00053  | 0.00000028|
| 5  | 0.90000| 0.89361| 0.88695    | 0.88680| 0.00015  | 0.00000002|
| 6  | 0.73317| 0.72756| 0.72773    | 0.72670| 0.00103  | 0.00000017|
| 7  | 0.61879| 0.64514| 0.66078    | 0.66020| 0.00058  | 0.00000033|
| 8  | 0.13067| 0.15270| 0.1618     | 0.16640| -0.00022 | 0.00000005|
| 9  | 0.78130| 0.78570| 0.78907    | 0.78810| 0.00097  | 0.00000094|
| 10 | 0.61931| 0.62311| 0.62665    | 0.62750| -0.00085 | 0.00000072|
| 11 | 0.46389| 0.46726| 0.47019    | 0.47060| -0.00041 | 0.00000016|
| 12 | 0.10000| 0.10492| 0.11106    | 0.11140| -0.00034 | 0.00000012|

Total SSE: 0.000000587
MSE: 0.000000049

### Table 4. Testing Model 3-10-1

| No | X1     | X2     | Target (Y) | Actual | Error   | SSE   | Performance |
|----|--------|--------|------------|--------|---------|-------|-------------|
| 1  | 0.81661| 0.81714| 0.81750    | 0.81680| 0.00070  | 0.00000049|
| 2  | 0.78208| 0.78386| 0.78466    | 0.78510| -0.00044 | 0.00000019|
| 3  | 0.57373| 0.57462| 0.57035    | 0.57750| -0.00715 | 0.00000510|
| 4  | 0.15073| 0.14174| 0.13507    | 0.14060| -0.00553 | 0.00000304|
| 5  | 0.90000| 0.89003| 0.87766    | 0.88180| -0.00414 | 0.00001713|
| 6  | 0.74648| 0.73731| 0.72610    | 0.73440| -0.00830 | 0.00000689|
| 7  | 0.64751| 0.64600| 0.63416    | 0.64770| -0.01354 | 0.00018322|
| 8  | 0.15812| 0.15820| 0.16150    | 0.16100| 0.00050  | 0.00000025|
| 9  | 0.80210| 0.80299| 0.80362    | 0.80330| 0.00032  | 0.00000010|
| 10 | 0.63541| 0.64075| 0.64671    | 0.64580| 0.00091  | 0.00000083|
| 11 | 0.47326| 0.47477| 0.47771    | 0.47740| 0.00031  | 0.00000009|
| 12 | 0.10000| 0.10231| 0.10650    | 0.10780| -0.00130 | 0.00000017|

Total SSE: 0.00035464
MSE: 0.00002955

b. Model 2-15-1
Figure 2. Results of Training Model 2-15-1

Figure 2 is the result of training with Matlab for model 2-15-1. The training of the model resulted in an Epoch of 217 iterations.

Table 5. Training Model 2-15-1

| No | X1 | X2 | Target (Y) | Actual | Error | SSE | Performance |
|----|----|----|------------|--------|-------|-----|-------------|
| 1  | 0.80177 | 0.80531 | 0.80618 | 0.80680 | -0.00062 | 0.00000039 |
| 2  | 0.76445 | 0.76911 | 0.77076 | 0.77210 | -0.00134 | 0.00000181 |
| 3  | 0.55719 | 0.56082 | 0.56531 | 0.56500 | 0.00031 | 0.00000010 |
| 4  | 0.14691 | 0.15581 | 0.16333 | 0.16280 | 0.00053 | 0.00000028 |
| 5  | 0.90000 | 0.89361 | 0.88695 | 0.88680 | 0.00015 | 0.00000002 |
| 6  | 0.73317 | 0.72756 | 0.72773 | 0.72660 | 0.00113 | 0.00000128 |
| 7  | 0.61879 | 0.64514 | 0.66078 | 0.66020 | 0.00058 | 0.00000033 |
| 8  | 0.13067 | 0.15270 | 0.16618 | 0.16630 | 0.00012 | 0.00000029 |
| 9  | 0.78130 | 0.78570 | 0.78907 | 0.78810 | 0.00097 | 0.00000094 |
| 10 | 0.61931 | 0.62311 | 0.62665 | 0.62740 | 0.00075 | 0.00000056 |
| 11 | 0.46389 | 0.46726 | 0.47019 | 0.47060 | 0.00041 | 0.00000016 |
| 12 | 0.10000 | 0.10492 | 0.11106 | 0.11140 | -0.00034 | 0.00000012 |

Total SSE 0.00000601
MSE 0.00000050

Table 6. Testing Model 2-15-1

| No | X1 | X2 | Target (Y) | Actual | Error | SSE | Performance |
|----|----|----|------------|--------|-------|-----|-------------|
| 1  | 0.81661 | 0.81714 | 0.81750 | 0.81680 | 0.00070 | 0.00000049 |
| 2  | 0.78208 | 0.78386 | 0.78466 | 0.78520 | -0.00054 | 0.00000029 |
| 3  | 0.57373 | 0.57462 | 0.57035 | 0.57750 | -0.00715 | 0.00005108 |
| 4  | 0.15073 | 0.14774 | 0.13507 | 0.14010 | -0.00503 | 0.00002535 |
| 5  | 0.90000 | 0.89003 | 0.87766 | 0.88190 | -0.00424 | 0.0001796 |
| 6  | 0.74648 | 0.73731 | 0.72610 | 0.73460 | -0.00850 | 0.00007227 |
| 7  | 0.64751 | 0.64600 | 0.63416 | 0.64770 | -0.01354 | 0.00018322 |
| 8  | 0.15812 | 0.15820 | 0.16150 | 0.16070 | 0.00080 | 0.00000645 |
| 9  | 0.80210 | 0.80299 | 0.80362 | 0.80330 | 0.00032 | 0.00000010 |
| 10 | 0.63541 | 0.64075 | 0.64671 | 0.64580 | 0.00091 | 0.00000083 |
| 11 | 0.47326 | 0.47477 | 0.47771 | 0.47730 | 0.00041 | 0.00000016 |
| 12 | 0.10000 | 0.10231 | 0.10650 | 0.10750 | -0.00100 | 0.00000101 |

Total SSE 0.00035342
MSE 0.00002945

c. Model 2-20-1
Figure 3 is the result of training with Matlab for model 2-20-1. The training of the model resulted in an Epoch of 153 iterations.

### Table 7. Training Model 2-20-1

| No | X1     | X2     | Target (Y) | Actual   | Error   | SSE         | Performance |
|----|--------|--------|------------|----------|---------|-------------|-------------|
| 1  | 0.80177| 0.80531| 0.80618    | 0.80640  | -0.00022| 0.000000005| 0.00000031  |
| 2  | 0.76445| 0.76911| 0.77076    | 0.77190  | -0.00114| 0.00000131  |
| 3  | 0.55719| 0.56082| 0.56531    | 0.56470  | 0.00061 | 0.00000038  |
| 4  | 0.14691| 0.15581| 0.16333    | 0.16320  | 0.00013 | 0.00000002  |
| 5  | 0.90000| 0.89361| 0.88695    | 0.88700  | -0.00005| 0.00000000  |
| 6  | 0.73317| 0.72756| 0.72773    | 0.72740  | 0.00033 | 0.00000011  |
| 7  | 0.61879| 0.64514| 0.66078    | 0.66060  | 0.00018 | 0.00000003  |
| 8  | 0.13067| 0.15270| 0.16618    | 0.16620  | -0.00002| 0.00000000  |
| 9  | 0.78130| 0.78570| 0.78907    | 0.78790  | 0.00117 | 0.00000013  |
| 10 | 0.61931| 0.62311| 0.62665    | 0.62730  | -0.00065| 0.00000042  |
| 11 | 0.46389| 0.46726| 0.47019    | 0.47040  | -0.00021| 0.00000004  |
| 12 | 0.10000| 0.10492| 0.11106    | 0.11110  | -0.00004| 0.00000000  |

Total SSE: 0.00000373
MSE: 0.00000031

### Table 8. Testing Model 2-20-1

| No | X1     | X2     | Target (Y) | Actual   | Error   | SSE         | Performance |
|----|--------|--------|------------|----------|---------|-------------|-------------|
| 1  | 0.81661| 0.81714| 0.81750    | 0.81680  | 0.00070 | 0.00000049  |
| 2  | 0.78208| 0.78386| 0.78466    | 0.78520  | -0.00054| 0.00000029  |
| 3  | 0.57373| 0.57462| 0.57035    | 0.57710  | -0.00675| 0.00004553  |
| 4  | 0.15073| 0.14174| 0.13507    | 0.14100  | -0.00593| 0.00003522  |
| 5  | 0.90000| 0.89003| 0.87766    | 0.88290  | -0.00524| 0.00002744  |
| 6  | 0.74648| 0.73371| 0.72610    | 0.73570  | -0.00960| 0.00009219  |
| 7  | 0.64751| 0.64600| 0.63416    | 0.64770  | -0.01354| 0.00018322  |
| 8  | 0.15812| 0.15820| 0.16150    | 0.16140  | 0.00010 | 0.00000001  |
| 9  | 0.80210| 0.80299| 0.80362    | 0.80340  | 0.00022 | 0.00000005  |
| 10 | 0.63541| 0.64075| 0.64671    | 0.64570  | 0.00101 | 0.00000103  |
| 11 | 0.47326| 0.47477| 0.47771    | 0.47690  | 0.00081 | 0.00000065  |
| 12 | 0.10000| 0.10231| 0.10650    | 0.10740  | -0.00990| 0.00000082  |

Total SSE: 0.00038693
MSE: 0.00003224
3.3. Best Model Selection and Performance
The following is a comparison table between the 2-10-1, 2-15-1 and 2-20-1 models.

| Bayesian Regularization | Information | Epoch | MSE Training | MSE Testing / Performance |
|-------------------------|-------------|-------|--------------|---------------------------|
|                         | 2-10-1      | 263   | 0.00000049   | 0.00002955               |
|                         | 2-15-1      | 217   | 0.00000050   | 0.00002945               |
|                         | 2-20-1      | 153   | 0.00000031   | 0.00003224               |

Based on the comparison, model 2-15-1 is the best model with an Epoch of 217, MSE training 0.00005050 and MSE Testing / Performance 0.00002945. This is because MSE Testing is more crisp than other models.

4. Conclusion
Bayesian Regularization method can be used to solve forecasting problems. This is because based on the results of the analysis, the error rate is quite low and the actual results are close to the desired target data. Based on a comparison of the 3 network architecture models used (2-10-1, 2-15-1 and 2-20-1), the 2-15-1 architectural model is the best model because the epoch is quite low and results in MSE testing and performance, which is better than other models.

References
[1] A. Wanto, M. Zarlis, Sawaluddin, and D. Hartama, “Analysis of Artificial Neural Network Backpropagation Using Conjugate Gradient Fletcher Reeves in the Predicting Process,” Journal of Physics: Conference Series, vol. 930, no. 1, pp. 1–7, 2017.
[2] A. Sagheer, M. Zidan, and M. M. Abdelsamea, “A Novel Autonomous Perceptron Model for Pattern Classification Applications,” Entropy, vol. 21, no. 8, pp. 1–24, 2019.
[3] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, “Backpropagation and the brain,” Nature Reviews Neuroscience, vol. 21, no. 6, pp. 335–346, 2020.
[4] Y. Y. Shen, Y. M. Zhang, X. Y. Zhang, and C. L. Liu, “Online semi-supervised learning with learning vector quantization,” Neurocomputing, vol. 399, pp. 467–478, 2020.
[5] S. Earp and A. Curtis, “Probabilistic neural network-based 2D travel-time tomography,” Neural Computing and Applications, vol. 32, no. 22, pp. 17077–17095, 2020.
[6] M. Kobayashi, “Hopfield neural networks using Klein four-group,” Neurocomputing, vol. 387, pp. 123–128, 2020.
[7] X. Xu and N. Gupta, “Application of radial basis neural network to transform viscoelastic to elastic properties for materials with multiple thermal transitions,” Journal of Materials Science, vol. 54, no. 11, pp. 8401–8413, 2019.
[8] T. Afriliansyah et al., “Implementation of Bayesian Regulation Algorithm for Estimation of Production Index Level Micro and Small Industry,” Journal of Physics: Conference Series, vol.
1255, no. 012027, pp. 1–6, Aug. 2019.

[9] M. M. Shora, H. Ghassemi, and H. Nowruzi, “Using computational fluid dynamic and artificial neural networks to predict the performance and cavitation volume of a propeller under different geometrical and physical characteristics,” *Journal of Marine Engineering and Technology*, vol. 17, no. 2, pp. 59–84, 2018.

[10] R. Mbuvha, M. Jonsson, N. Ehn, and P. Herman, “Bayesian neural networks for one-hour ahead wind power forecasting,” in *6th International Conference on Renewable Energy Research and Applications, ICRERA*, 2017, pp. 591–596.

[11] E. Lázaro, C. Armero, and D. Alvares, “Bayesian regularization for flexible baseline hazard functions in Cox survival models,” *Biometrical Journal*, vol. 63, no. 1, pp. 7–26, 2021.

[12] M. Byrd, L. H. Nghiem, and M. McGee, “Bayesian Regularization of Gaussian Graphical Models With Measurement Error,” *Computational Statistics and Data Analysis*, vol. 156, no. 107085, pp. 1–20, 2020.

[13] L. Gan, N. N. Narisetty, and F. Liang, “Bayesian Regularization for Graphical Models With Unequal Shrinkage,” *Journal of the American Statistical Association*, vol. 114, no. 527, pp. 1218–1231, 2019.

[14] D. Hu, J. Zhang, and Z. W. Liu, “Charging stations expansion planning under government policy driven based on Bayesian regularization backpropagation learning,” *Neurocomputing*, vol. 416, pp. 47–58, 2020.

[15] I. Khan et al., “Design of Neural Network with Levenberg-Marquardt and Bayesian Regularization Backpropagation for Solving Pantograph Delay Differential Equations,” *IEEE Access*, vol. 8, no. 0, pp. 137918–137933, 2020.

[16] E. Sariev and G. Germano, “Bayesian regularized artificial neural networks for the estimation of the probability of default,” *Quantitative Finance*, vol. 20, no. 2, pp. 311–328, 2020.

[17] A. Wanto and J. T. Hardinata, “Estimations of Indonesian poor people as poverty reduction efforts facing industrial revolution 4.0,” *IOP Conference Series: Materials Science and Engineering*, vol. 725, no. 1, pp. 1–8, 2020.

[18] A. Wanto et al., “Forecasting the Export and Import Volume of Crude Oil, Oil Products and Gas Using ANN,” *Journal of Physics: Conference Series*, vol. 1255, no. 1, pp. 1–6, 2019.

[19] A. Wanto et al., “Analysis of the Backpropagation Algorithm in Viewing Import Value Development Levels Based on Main Country of Origin,” *Journal of Physics: Conference Series*, vol. 1255, no. 1, pp. 1–6, 2019.

[20] G. W. Bhawika et al., “Implementation of ANN for Predicting the Percentage of Illiteracy in Indonesia by Age Group,” *Journal of Physics: Conference Series*, vol. 1255, no. 1, pp. 1–6, 2019.

[21] N. L. W. S. R. Ginantra, M. A. Hanafiah, A. Wanto, R. Winanjaya, and H. Okprana, “Utilization of the Batch Training Method for Predicting Natural Disasters and Their Impacts,” *IOP Conf. Series: Materials Science and Engineering*, vol. 1071, no. 012022, pp. 1–7, 2021.