Journal of Theoretical and Applied Information Technology
31st August 2023. Vol.101. No 16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NOVEL METHOD TO ESTIMATE HIDDEN NEURONS IN ENSEMBLE FLOOD FORECASTING MODEL

NAZLI MOHD KHAIRUDIN [1], NORWATI MUSTAPHA [2], TEH NORANIS MOHD ARIS [3], MASLINA ZOLKEPLI [4]

[1,2,3,4]Universiti Putra Malaysia, Faculty of Computer Sciences and Information Technology, Malaysia

Email: [1]nazmkhair@gmail.com, [2]norwati@upm.edu.my, [3]nuranis@upm.edu.my, [4]masz@upm.edu.my

ABSTRACT

Machine learning model such as neural networks have been widely adopted to provide flood forecast. Self-adaptability in neural networks enable them to learn pattern by their own and adjusted the connection between the neurons, but still producing error. In many neural network adaptations for flood forecasting, the number of hidden neurons is normally randomly selected which can caused overfitting problem in the network. In this study, a novel method to estimates the hidden neuron is proposed to overcome this problem. This method integrates the evaluation of various convergence theorem criteria with grid search to estimates the hidden neuron. By having this integration, optimal number of fix hidden neurons can be determined. This method is used in the ensemble model that based on neural networks to forecast the water level based on rainfall data. Based on the performance measurement using Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Nash Sutcliffe Efficiency (NSE), it is found that the integration of convergence theorem and grid search can be used to fix the number of hidden neuron and has reduce the error that led to overfitting in the forecast.

Keywords: *Neural network, Hidden neuron estimation, Flood forecasting, Grid search, Convergence Theorem*

1. INTRODUCTION

Machine learning model plays an important role in the dynamic nature of environment by the development of many forecast model for strategic move and risk management. Flood in recent context has been occurred around the world that impacted life in a more tremendous way. Flood caused by the heavy rainfall is potential to be forecast earlier in long-term manner. Due to the non-linearity and dynamic nature of rainfall the result of such forecast model would be varies with high error, thus an accurate forecast is needed.

In the adoption of machine learning nowadays to provide flood forecast, it can be categorized as single or hybrid model. Single model only applied one machine learning approach to provide the forecast, whereas hybrid model may have more than one model or approach through integration, combination or ensemble of models [1]. Although single model can provide a satisfactory forecast but it is found that the hybridization of models can improve the performance of a single model forecast [2]. Neural networks such as ANN has been widely popular among researchers either to be used in single model [3] or apart of hybrid model [4]-[5]. This is due to its self-adaptability that can learn pattern from a given series of dataset [6].

Neural network is a system modelled to imitate how the human learns by recognizing the pattern based on the set of input. It is made up of three connected layers named input, hidden, and output layers that adaptively learn the underlying relationship of the data to generate the desired output [7]. It can learn by their own or example and make the necessary adjustment to the connection between the neurons. Designing neural networks come with a challenge of designing the training process.

During the training process, the aim of the network is reducing the error of the output [8]. In many adaptations of neural network today, the random assignment of hidden neuron can lead to high variance in the network and can cause overfitting or underfitting problem [9].

Overfitting happened when the network closely matches the dataset and lose its ability to generalize over the test data. While underfitting make it hard to apprehend the relationship between input and output. As neural network aiming to produce a very small error, the selection of the optimal hidden neuron number is very important [10]. The error produce by the nodes in the network can be influenced by the hidden neurons in which their output is connected. It is found that better stability of the network can be achieved with minimal error, while higher error depicts worst stability.

Researchers in recent times have implemented and proposed several methods in estimating the number of hidden neurons in hydrological forecasting [11]–[13]. Estimating of hidden neurons with trial-and-error approach is the most popular method. This method adjusted the complexity of the networks with additional neurons until there is no improvement in the results [14]. The trial-and-error approach has been used to forecast streamflow [7], rainfall [15], discharge [12], and river level [16]. Although trial-and-error approach provide an efficient way to provide optimal number of hidden neurons, but it is hard to determine the initial number of hidden neurons to be considered for trial since there is no basis knowledge of the problem under consideration. Furthermore, adding more nodes to the networks can lead to overfitting, while insufficient hidden node can lead the model to be invalid [6].

A study by [17] used a more systematically method of fixing hidden neuron by using grid search. Grid search has overcome the overfitting problem with determining the optimal hidden neuron by analysing different configurations of the network elements. Although it can be a very exhaustive process when various hyperparameters are used, but it provides a basis on the initial configurations and the complexity of the networks can be estimated.

Estimating the number of hidden neurons also can be done by analysing the empirical relationship between input and output. A study by [13] used the empirical equation of $2n+1$ to find optimal hidden neuron in forecasting the water level of Klang River. From the equation, n represent the number of the input neurons. By having an optimal hidden neuron value, the forecast has been in satisfactory whereby the model able to forecast near the observed value. It is also mentioned by [6] that number of neuron in hidden layer can be ranged from $(2n+1)$ to $(2\sqrt{n}+m)$, where n and m represent the number of input neuron and output neuron respectively.

Another empirical equation used by [12] to estimate hidden neuron is $N = \sqrt{E + S} + a$, where N represent the number of hidden neurons, E represent the input variables ($E \in [1;10] \cap \mathbb{N}$), S represent the number of output neurons ($S = 1$) and a is adjustment parameter ($a \in [1;10] \cap \mathbb{N}$), therefore $N \in [2;14] \cap \mathbb{N}$. The optimal number of neurons has been achieved by this equation and used to forecast daily discharge of river in South France. It was shown that the training process has improved and forecast gain satisfactory. It is proven that the number of hidden neurons can be estimated well by empirical analysis. However, the extensive knowledge was still needed to find the suitable equation for domain and problem under research.

Ant colony optimization (ACO) has also been explored in finding optimal number of hidden neuron to be used in the rainfall forecast [11]. Initially, each ant is initialized with various hidden node and pheromones is set to zero. Root Mean Squared Error (RMSE) is calculated for each ant fitness and pheromones are updated in each iteration along with the global best until the number of hidden neurons is determined for the model. Although the estimation of the hidden neurons number in the rainfall forecast model has improved the model performance, but ACO is known to be trap in local minima and have slow convergence rate [18]. This may lead to uncertainty in the hidden neuron's estimation.

Despite the good performance offered by the existing methods in estimating hidden neuron, there are still some shortcomings that need to be improved in minimizing overfitting problem. Motivate by the recent ideas, a novel method to estimate the hidden neurons number in flood forecasting is proposed based on the criteria that fit the convergence theorem and grid search. The convergence criteria converge the infinite sequences to finite sequences, while grid search find the optimum value of hidden neuron. The convergence criteria provide a based initialization of the number of hidden neurons and grid search provides a best way to determine the optimal setting for the hyperparameters.

Journal of Theoretical and Applied Information Technology
31st August 2023. Vol.101. No 16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The convergence theorem allows systematic exploration of different hyperparameter configurations, such as the number of hidden neurons, when applied to neural networks and grid search. Grid search performs a thorough search of the parameter space while training numerous models using various hidden neuron values. The convergence theorem guarantees that when additional data is utilized for training, the models will become closer to the actual underlying patterns, producing precise estimates of the ideal number of hidden neurons. Because the neural network may use the appropriate level of complexity to capture complicated correlations in the data, improving its predictive powers, this method helps to decrease forecast mistakes.

2. NOVEL METHODS TO ESTIMATES HIDDEN NEURON IN NEURAL NETWORKS

In training the neural network model, there is an issue of overfitting. Overfitting will result from too many hidden neurons in which the neural networks overestimated the difficulty of the target problem [19]. Overfitting may lose the model generalization ability and providing an accurate result. To avoid over fitting, it is crucial to estimates the right number of hidden neurons for the network.

In designing the neural network process, a novel method is proposed to estimate the number of hidden neurons by statistical criteria that fit the convergence theorem. The optimal value of the hidden neuron is determined by grid search process. The flow of the proposed method is as in Figure 1.

The interaction between model complexity, overfitting, and generalization serves as the theoretical underpinning for applying the convergence theorem with grid search to set the number of hidden nodes in neural networks. Because of the convergence theorem, forecast errors are minimized as the neural network's predictions get closer to the actual underlying patterns in the data as training data volume rises.

Grid search adds to this by methodically investigating different hyperparameter setups, such as the quantity of hidden nodes. Grid search tries to find the best compromise between model complexity and generalization by training and evaluating models with various hidden node values, assisting in the identification of the appropriate number of hidden nodes that may successfully capture the underlying patterns in the data without overfitting. The neural network's prediction abilities are improved and its performance on new data is enhanced by this combination method.

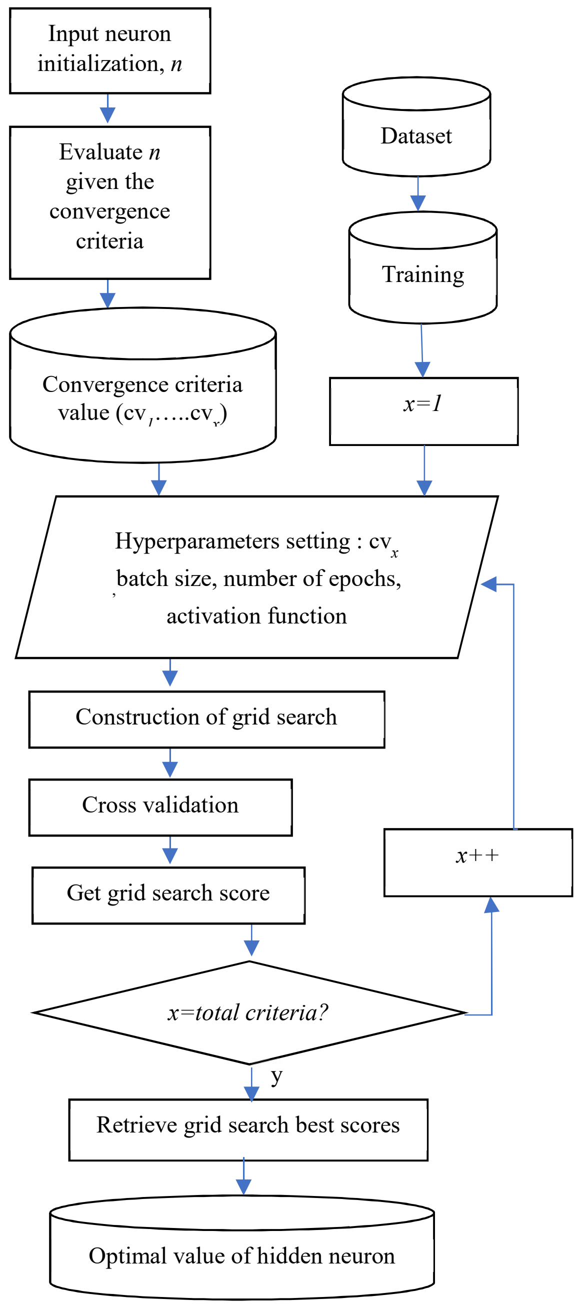


Figure 1: The proposed hidden neuron estimation method

The number of hidden nodes in neural networks can be fixed using the convergence theorem and grid search, which will reduce overfitting and improve prediction accuracy. As more data is utilized for training, the convergence theorem guarantees that the neural network's predictions will move closer to the actual underlying patterns, improving generalization. Grid search methodically investigates various hyperparameter settings, determining the ideal number of hidden nodes that strikes a balance between model complexity and generalization, effectively limiting overfitting and producing enhanced forecast accuracy on unobserved data.

Any dataset that been used for the model is divided into training and testing. The number of input neuron then be determined from the dataset. The proposed method examines various criteria that fit the convergence theorem in estimating the training process of the neural networks. These criteria were introduced by [20] and never been applied in flood forecasting. The criterion under evaluation is as in Table 1. For every criterion, the input neuron is taken into consideration and examined using the convergence theorem. Convergence converts an infinite sequence to a finite one. For input neuron n, the corresponding value of n for each criterion is $cv_1 \ldots\ldots cv_x$ where x is the number of total criteria.

Table 1: Criteria that fit convergence theorem [20]

No.	Criteria	No.	Criteria
1	(10n+1)/n	50	$5(n^2+6)-2/n^2-8$
2	(11n+1)/n	51	$5(n^2+7)/n^2-8$
3	(9n+1)/n	52	$5(n^2+8)/n^2-8$
4	2n/(n+1)	53	$5(n^2+8)+1/n^2-8$
5	2n/(n-1)	54	$5(n^2+9)/n^2-8$
6	3(n+1)/n	55	$5(n^2+9)-1/n^2-8$
7	3n/(n-1)	56	$5(n^2+9)-2/n^2-8$
8	3n+5/n-2	57	5n/(n-1)
9	3n+7/n-2	58	5n/(n-2)
10	3n+8/n-2	59	5n+4/n-2
11	$3n^2+7/n^2-8$	60	5n+5/n-2
12	$3n^2+8/n^2-8$	61	5n+6/n-2
13	4.5n/(n-1)	62	5n+7/n-2
14	4n/(n-1)	63	$5n^2/n^2-8$
15	4n/n-2	64	$5n^2+1/n^2-8$
16	4n+1/n-2	65	$5n^2+2/n^2-8$
17	$4n^2+1/n^2-8$	66	$5n^2+3/n^2-8$
18	$4n^2+2/n^2-8$	67	$5n^2+4/n^2-8$
19	$4n^2+3/n^2-8$	68	$6(n^2+4)/n^2-8$
20	$4n^2+4/n^2-8$	69	$6(n^2+4)+2/n^2-8$
21	$4n^2+5/n^2-8$	70	$6(n^2+4)+3/n^2-8$
22	$4n^2+6/n^2-8$	71	$6(n^2+5)/n^2-8$
23	$4n^2+7/n^2-8$	72	$6(n^2+5)-1/n^2-8$
24	$4n^2+8/n^2-8$	73	$6(n^2+7)+2/n^2-8$
25	$5(n^2+1)/n^2-8$	74	6n/(n-2)
26	$5(n^2+1)+1/n^2-8$	75	$7(n^2+2)/n^2-8$
27	$5(n^2+1)+2/n^2-8$	76	$7(n^2+4)/n^2-8$
28	$5(n^2+1)+3/n^2-8$	77	$7(n^2+4)+1/n^2-8$
29	$5(n^2+1)+4/n^2-8$	78	$7(n^2+4)+2/n^2-8$
30	$5(n^2+2)/n^2-8$	79	$7(n^2+4)+3/n^2-8$
31	$5(n^2+2)+1/n^2-8$	80	$7(n^2+5)+1/n^2-9$
32	$5(n^2+2)+2/n^2-8$	81	$7(n^2+5)+2/n^2-8$
33	$5(n^2+2)+3/n^2-8$	82	$7(n^2+5)+3/n^2-8$
34	$5(n^2+2)+4/n^2-8$	83	$7(n^2+5)-1/n^2-8$
35	$5(n^2+2)-1/n^2-8$	84	$7(n^2+5)-2/n^2-8$
36	$5(n^2+3)/n^2-8$	85	$7(n^2+5)-3/n^2-8$
37	$5(n^2+3)+2/n^2-8$	86	7n+2/(n-2)
38	$5(n^2+3)+3/n^2-8$	87	$7n^2+13/n^2-8$
39	$5(n^2+4)/n^2-8$	88	8n/(n-2)
40	$5(n^2+4)+1/n^2-8$	89	8n+1/n-2
41	$5(n^2+4)+2/n^2-8$	90	8n+2/n-2
42	$5(n^2+4)+3/n^2-9$	91	8n+5/n-2
43	$5(n^2+4)-1/n^2-8$	92	8n+6/n-2
44	$5(n^2+4)-3/n^2-8$	93	$8n^2/n^2-7$
45	$5(n^2+5)/n^2-8$	94	9n/(n-2)
46	$5(n^2+5)+1/n^2-8$	95	9n+1/n-2
47	$5(n^2+5)+2/n^2-8$	96	9n+6/n-2
48	$5(n^2+6)+3/n^2-8$	97	n/(n+1)
49	$5(n^2+6)-1/n^2-8$		

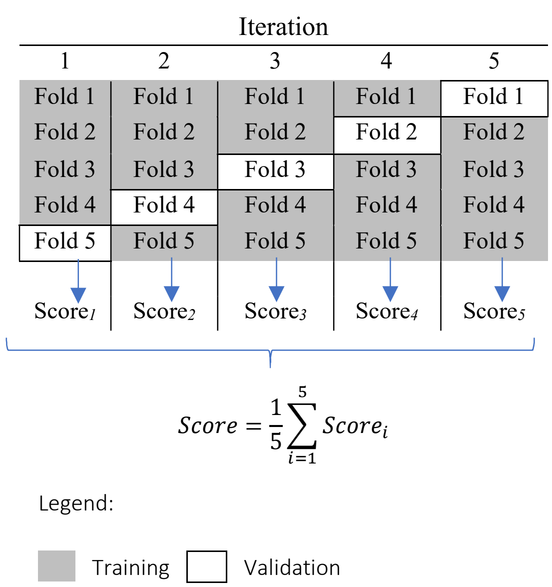
Once all criteria has been evaluated, the criteria's values along with other hyperparameters including batch size, number of epochs and activation function are tuned using the grid search space approach to find the best model [21]. In this approach, learning models are construct using various setting of hyperparameters of the training dataset. Cross validation procedure using *k*-fold is

Journal of Theoretical and Applied Information Technology
31st August 2023. Vol.101. No 16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

used to assess the performance of the constructed model. Cross validation works by dividing the train data into *k* equal parts. In this study, 5-fold cross validation is used as in Figure 2. The model is fit with *k-1* fold of the training data, and it is validated with the last block. The process is repeated for *k* times, and the mean cross-validated score of the best estimator gain from the repeated iteration is then averaged to evaluate the model.

The grid search with cross validation is done for all tuned hyperparameters that includes the criteria values that fit the convergence theorem. The model with the highest score is naturally considered to be the most optimal. Model that performs the best with the tuned hyperparameters are selected as the final model. This final model is the one that been evaluated using the testing set. The tuning of hyperparameters in learning network do have significant impact towards the model [22]. The algorithm for the proposed hidden neuron estimation can be present as Algorithm 1.



$$Score = \frac{1}{5}\sum_{i=1}^{5} Score_i$$

Figure 2: k-fold cross validation, k=5

Algorithm 1:

1. Initialize the number of input neuron, *n*.
2. Evaluate *n* for each of criteria that fit the convergence theorem.
3. Get the value for each of the criteria evaluation, $cv_1 \ldots cv_x$; where *x* is the number of total criteria.
4. Prepare the training data.
5. Initialize *x*=1.
6. Construct grid search space with hyperparameters: criteria value (cv_x), batch size, number of epochs and activation function.
7. Cross validates the model with hyperparameters in step 6 using *k-fold*.
    a. Calculate the cross-validation score for each iteration of the *k-fold*.
    b. Find mean score of all cross-validation score generated from all iteration.
8. Is all criteria value has been trained? If yes, go to step 9, if no repeat step 6-8 until all criteria value has been trained.
9. Retrieve the best score from the cross-validation process.
10. Get the optimal value of hidden neuron based on the best score.

3. EXPERIMENTS

To demonstrate the proposed method of estimating the hidden neuron in neural network, an ensemble neural network model is developed to forecast water level based on the real-time rainfall data.

3.1 Dataset

Monthly rainfall has been collected from Kelantan River Basin to forecast the water level of Kuala Krai. There are sourced from eight rainfall station along the Lebir River and Galas River which are Gunung Gagau, Kuala Koh, Kampung Aring, Kampung Lalok, Kampung Tualang and Kuala Krai, Dabong and Limau Kasturi. These stations located at the upstream while the water level station of Kuala Krai located at the downstream.

Forecasting the water level of Kuala Krai is utmost important as it is always faced risk to be flooded especially at the end of the year due to its surrounding that exposed to the north-east monsoon. Dataset from 2011 to 2019 are gathered from all stations. The dataset is divided into training and testing with the proportion of 75% and 25% respectively. The

Journal of Theoretical and Applied Information Technology
31st August 2023. Vol.101. No 16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

dataset can be described as in Table 2. The total records for each of the stations is 104.

Table 2: Dataset Description

	Mean	Std Dev	Min	Max
Input (Rainfall Stations)				
Gunung Gagau	302.32	342.49	0	2634
Kuala Koh	202.35	263.84	0	2094
Kampung Aring	202.13	148.66	0	1036.5
Tualang	204.34	242.86	0	1780.9
Kampung Lalok	198.22	168.98	0.2	1254.5
Kuala Krai	184.81	160.33	0	842.2
Limau Kasturi	171.76	190.94	0	1229
Dabong	198.72	156.83	0	1171
Output (Water Level Station)				
Kuala Krai	16.72	1.19	13.73	22.12

**** Std Dev = Standard Deviation***

3.2 Performance Measurement

The estimation of hidden neuron aims to produce output with minimal error. Thus, to validate the proposed method, statistical method is used to evaluate the performance of the model with estimated optimal hidden neuron. Often known as the "goodness of fit", model result is evaluated against the observed value by measurements below:

i. Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2} \quad (1)$$

ii. Mean Absolute Error (MAE)

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}| \quad (2)$$

where, y_i is the original value at period i, $\hat{y}_i$ is the forecasted value at the period of i, and N denotes the number of the sample.

iii. Nash-Sutcliffe Efficiency (NSE)

$$NSE = 1 - \frac{\sum_{t=1}^{T}(Q_m^t - Q_0^t)^2}{\sum_{t=1}^{T}(Q_o^t - \bar{Q}_0)^2} \quad (3)$$

where, Q_0 is the original value, and Q_m is the forecasted value, while Q^t_0 is the original value at time t.

In Eq. (1) and Eq. (2), y_i is the original value at period i, $\hat{y}_i$ is the forecasted value at the period of i, and N denotes the number of the sample. While in (3), Q_0 is the original value, and Q_m is the forecasted value. Q_o^t is the original value at time t. The smallest number produce by Eq. (1) and Eq. (2) denote best performances while highest number produce by Eq. (3) denotes better forecast performance.

These performance measurements are applicable to be used in various machine learning, thus appropriate for this study. It has been used in many hydrological forecasting such as ground water forecast [23], runoff forecast [24], water level forecast [25], and flow forecast [26].

3.3 Model Development

To verify the proposed estimation hidden neuron method, an ensemble neural network is developed based on two neural networks which are Backpropagation Neural Networks (BPNN) and Extreme Learning Machine (ELM). These neural networks are trained independently, and the forecast result are then averaged to produce the final forecast. The hidden neuron estimation is only done for the BPNN network. Figure 3 presents the flow of the ensemble model with hidden neuron estimation while Table 3 describes the design parameters of the BPNN.

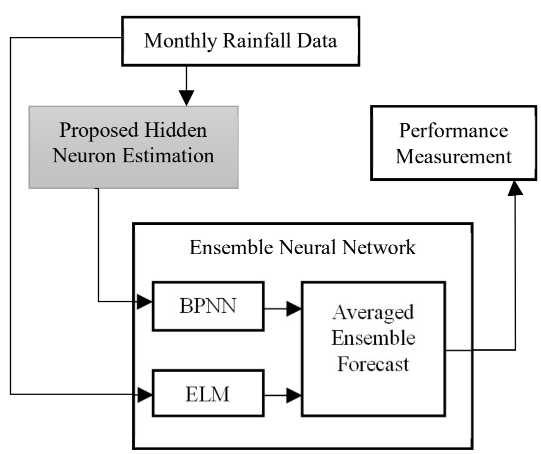


Figure 3: Ensemble Model with proposed hidden neuron estimation method

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3: BPNN model design parameters

Hyperparameters	Value
Output neuron	1
Number of hidden layers	2
Input neurons	8
Number of epochs	100
Activation function evaluated	"sigmoid", "relu"
Number of convergence criterion	101
Number of batches	32

Combining the results from different neural networks model can add bias that balance the variance of independent model. It has the advantages in which it is not fully dependent on the input data and training scheme.

2. RESULTS AND DISCUSSION

The forecast performance of the ensemble neural networks model with the proposed hidden neuron estimation using criteria that fit the convergence theorem is statistically measured using RMSE, MAE and NSE. To validate the results, the ensemble forecast model results is compared against ensemble model that used the existing empirical equation of (2n+1) and N = √ E + S + a, in estimating the hidden neuron in flood forecasting. Table 4 summarize the results of all ensemble model performance using existing and proposed method.

Table 4: Ensemble Model Performance

References	[6], [13]	[12]	Proposed method
Method	N=2n+1	N = √ (E + S) + a	N=(8*n) + 6 /(n-2)
Number of Hidden Neuron	17	11	65
RMSE	1.3595	1.4051	**1.3353**
MAE	0.8639	0.9276	**0.8433**
NSE	-0.7739	-2.9504	**-0.5184**

Based on the proposed hidden neuron estimation method, the number of optimal hidden neuron produced by the grid search is 65 with the best activation function using "relu". The criteria are set with eight input neurons. It is found that the best criteria for the proposed hidden neuron estimation, N is as follows:

$$N=8n + 6 /(n-2) \qquad (4)$$

where n is the number of input neuron. Although the proposed criteria have provided better accuracy for the ensemble forecast, but grid search involved a time-consuming process per execution.

By using the selected optimal hidden neuron in the ensemble model, it has indicated that it outperformed ensemble model with existing criteria based on RMSE, MAE and NSE value. The ensemble model with the proposed method has achieved minimal error of RMSE and MSE and higher NSE compared to existing approaches from the literature. Although the number of hidden neurons produce by the proposed method seem to be higher that existing method, but it has shown the reduction of error and increase the model energy.

Table 5a showing the observed value of water level compared to the forecasted value for the training data while Table 5b for the testing data.

Table 5a: Observed Water Level Compared to Forecasted Water Level for Training Data

Month	Observed Value	Forecasted Value
1	16.80	16.30
2	16.66	16.48
3	16.68	16.24
4	16.30	16.47
5	16.33	16.59
6	16.83	16.06
7	17.05	16.80
8	18.20	18.57
9	18.14	18.18
10	18.72	18.13
11	16.80	15.95
12	16.71	16.19
13	17.12	16.57
14	16.72	16.61
15	15.88	15.90
16	15.82	16.22
17	16.04	16.30
18	16.28	16.44
19	16.21	16.02
20	16.35	15.99
21	17.46	18.96

Month	Observed Value	Forecasted Value
22	17.58	16.57
23	17.59	18.91
24	16.63	15.87
25	16.17	16.54
26	16.23	16.43
27	15.81	16.06
28	16.06	16.11
29	16.51	16.86
30	16.50	16.56
31	17.03	17.14
32	17.37	17.26
33	18.94	19.97
34	17.32	16.73
35	15.74	15.56
36	15.51	15.89
37	15.62	15.79
38	15.93	16.10
39	15.96	16.02
40	15.70	15.84
41	15.99	16.14
42	16.37	16.26
43	16.96	16.63
44	17.14	17.65
45	22.12	23.79
46	18.56	16.37
47	17.01	15.67
48	15.77	15.65
49	15.54	15.89
50	16.13	16.46
51	15.87	15.95
52	15.48	16.23
53	16.22	16.93
54	16.34	16.23
55	16.50	16.31
56	17.39	16.81
57	17.62	17.64
58	17.30	16.11
59	16.94	16.40
60	15.67	15.55
61	15.33	15.58
62	15.36	16.10
63	15.98	16.12
64	15.88	16.19
65	15.66	16.22
66	15.75	16.21
67	15.61	16.06
68	16.45	16.36
69	17.46	17.99
70	20.06	20.36
71	17.40	16.19
72	16.72	15.99
73	16.96	15.97
74	17.40	17.04
75	17.11	16.92
76	17.05	16.32
77	17.16	16.62
78	17.47	16.40

Table 5b: Observed Water Level Compared to Forecasted Water Level for Testing Data

Month	Observed Value	Forecasted Value
79	17.28	16.79
80	18.52	19.14
81	18.36	18.83
82	19.08	19.25
83	16.72	15.94
84	16.12	16.24
85	15.94	16.16
86	15.99	16.77
87	16.12	16.23
88	16.14	16.24
89	16.45	16.04
90	16.33	16.29
91	16.35	16.77
92	16.24	16.60
93	17.70	19.25
94	13.73	17.16
95	16.02	15.90
96	15.05	15.78
97	15.09	16.15
98	15.43	16.03
99	15.82	16.24
100	15.77	15.95
101	15.96	16.07
102	17.52	15.89
103	19.09	15.94
104	20.66	16.83

The trend between the forecast water level and observed water level in the training and testing phase can be shown in Figure 4.

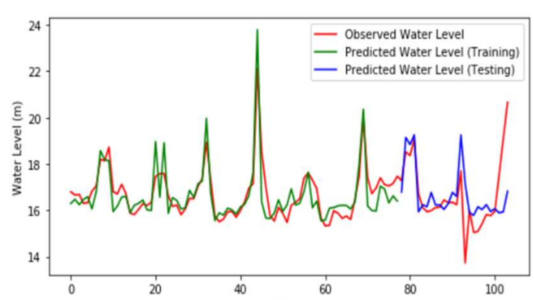


Figure 4: Forecasted Water Level vs Observed Water Level

From the pattern of the ensemble model forecast, it is proved that the forecasted water level has the best agreement with the observed water level, thus reducing the high variance of the network that led to overfitting.

Integrating the convergence theorem with grid search to fix hidden nodes in neural networks typically beats both trial-and-error and ant colony optimization methods in terms of accuracy and error rate. The number of hidden nodes is one of many hyperparameter settings that are carefully explored through the integration of convergence theorem and grid search. By comparing many models, it can determine the best compromise between model complexity and generalization, improving accuracy and reducing error rates on unobserved data.

This has advantages over the manual selection of hyperparameters in the trial-and-error method, which can be time-consuming and ineffective. It might not produce the best-performing model because the approach doesn't systematically explore the hyperparameter space, which could lead to less accurate results and higher error rates. Comparatively speaking, Ant Colony Optimization (ACO) may suffer in the high-dimensional and complex search space of neural networks, while being a potent optimization approach. It might be computationally expensive and still be subject to local optima, which would restrict its capacity to achieve high precision and low error rates.

The convergence criteria being used in the ensemble model is all satisfied with the convergence theorem. The important aspect of convergence is the sequence has to be finite limit [20]. The proof that the proposed criteria satisfy the convergence theorem as below:

$$N=8n + 6 / (n-2),$$

$$\lim_{n\to\infty} \frac{8n+6}{n-2} = \frac{8n(1+\frac{6}{8n})}{n(1-\frac{2}{n})} = \frac{8(1+0)}{1-0} = 8 \quad (5)$$

where n is the number of input neurons, and the equation results is a finite value. It is proved that the proposed criteria have converged with a bounded sequences and limit value.

5. CONCLUSION

Various research has proposed several approaches in determining the number of hidden neurons in flood forecasting, such as trial-and-error and empirical equations to reduce error and improve forecast accuracy. However, they still have limitations. A novel method to estimates the hidden neuron number is proposed based on the criteria that fit the convergence theorem. It has been found that the criterion which satisfying the convergence theorem under study is 8n + 6 /(n-2) where n is the number of input neuron. This criterion is then validated by applying it into ensemble neural network model to forecast water level based on real-time rainfall data collected from Kelantan River Basin.

Based on the performance measurement of RMSE, MAE and NSE of the ensemble model, it is found that network error is reduced using the proposed criteria compared to the existing criteria of estimating hidden neurons. The model with minimum error reduces the variance of the network, thus reducing the overfitting in training the dataset.

The convergence theorem guarantees that the neural network's predictions get closer to actual patterns in the data as more data is utilized for training. As the model improves at generalizing to new cases, it helps to reduce overfitting. Grid search offers a methodical way to investigate various hyperparameter configurations and makes it easier to find the best value for the hidden neurons, which is otherwise difficult to do through trial and error.

The convergence theorem and grid search are combined to identify the ideal number of hidden neurons, resulting in a better compromise between model complexity and generalization. This improves the neural network's capacity to anticipate new input accurately. Convergence theorem-driven training and systematic hyperparameter exploration improve model performance by lowering forecast errors and raising the overall accuracy of the ensemble neural network.

The convergence theorem and grid search can be used for a variety of neural network tasks and topologies. It makes that the model is tuned

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

for each unique dataset, improving generalization to a wider range of scenarios. Convergence theory and grid search integration in resolving hidden neurons provides comprehensive answers to key issues such hyperparameter tuning, overfitting, generalization, and model performance. It offers a sensible and effective method for optimizing neural networks, resulting in greater precision and more trustworthy predictions for practical applications.

REFERENCES

[1] N. M. Khairudin, N. Mustapha, T. N. M. Aris, and M. Zolkepli, "in-Depth Review on Machine Learning Models for Long-Term Flood Forecasting," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 10, pp. 3360–3378, 2022.

[2] A. Mosavi, P. Ozturk, and K. W. Chau, "Flood prediction using machine learning models: Literature review," *Water (Switzerland)*, vol. 10, no. 11, pp. 1–40, 2018, doi: 10.3390/w10111536.

[3] A. R. Sanubari, P. D. Kusuma, and C. Setianingsih, "Flood modelling and prediction using artificial neural network," *Proceedings - 2018 IEEE International Conference on Internet of Things and Intelligence System, IOTAIS 2018*. pp. 227–233, 2019, doi: 10.1109/IOTAIS.2018.8600869.

[4] B. Yaghoubi, S. A. Hosseini, and S. Nazif, "Monthly prediction of streamflow using data-driven models," *J. Earth Syst. Sci.*, vol. 128, no. 6, pp. 1–15, 2019, doi: 10.1007/s12040-019-1170-1.

[5] Z. Zahmatkesh and E. Goharian, "Comparing machine learning and decision making approaches to forecast long lead monthly rainfall: The city of Vancouver, Canada," *Hydrology*, vol. 5, no. 1, pp. 1–22, 2018, doi: 10.3390/hydrology5010010.

[6] A. Hawamdeh and M. Al Kuisi, "An artificial neural network model for flood forecasting, case study in Jordan," *Solid State Technol.*, vol. 64, no. 2, pp. 4704–4714, 2021.

[7] S. Agarwal, P. J. Roy, P. Choudhury, and N. Debbarma, "Flood Forecasting and Flood Flow Modeling in a River System Using ANN," *Water Pract. Technol.*, vol. 16, no. 4, pp. 1194–1205, 2021, doi: 10.2166/wpt.2021.068.

[8] P. Mitra *et al.*, "Flood forecasting using Internet of things and artificial neural networks," *7th IEEE Annu. Inf. Technol. Electron. Mob. Commun. Conf. IEEE IEMCON 2016*, pp. 1–5, 2016, doi: 10.1109/IEMCON.2016.7746363.

[9] T. Zhou, F. Wang, and Z. Yang, "Comparative analysis of ANN and SVM models combined with wavelet preprocess for groundwater depth prediction," *Water (Switzerland)*, vol. 9, no. 10, 2017, doi: 10.3390/w9100781.

[10] J. Li, Y. Wu, J. Zhang, and G. Zhao, "A Novel Method to Fix Numbers of Hidden Neurons in Deep Neural Networks," *2015 8th Int. Symp. Comput. Intell. Des.*, vol. 2, 2015, doi: 10.1109/ISCID.2015.41.

[11] K. Varada Rajkumar and D. K. Subrahmanyam, "A Novel Method for Rainfall Prediction and Classification using Neural Networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 7, pp. 521–528, 2021, doi: 10.14569/IJACSA.2021.0120760.

[12] F. Y. Dtissibe, A. A. A. Ari, C. Titouna, O. Thiare, and A. M. Gueroui, "Flood forecasting based on an artificial neural network scheme," *Nat. Hazards*, vol. 104, no. 2, pp. 1211–1237, 2020, doi: 10.1007/s11069-020-04211-5.

[13] J. L. Hong and K. Hong, "Flood Forecasting for Klang River at Kuala Lumpur using Artificial Neural Networks," *Int. J. Hybrid Inf. Technol.*, vol. 9, no. 3, pp. 39–60, 2016, doi: 10.14257/ijhit.2016.9.3.05.

[14] S. Dazzi, R. Vacondio, and P. Mignosa, "Flood stage forecasting using machine-learning methods: A case study on the parma river (Italy)," *Water (Switzerland)*, vol. 13, no. 12, 2021, doi: 10.3390/w13121612.

[15] S. V. Lakshminarayana, "Rainfall Forecasting using Artificial Neural Networks (ANNs): A Comprehensive Literature Review," *Indian J. Pure Appl. Biosci.*, vol. 8, no. 4, pp. 589–599, 2020, doi: 10.18782/2582-2845.8250.

[16] K. C. Keong, M. Mustafa, A. J. Mohammad, M. H. Sulaiman, and N. R. H. Abdullah, "Artificial neural network flood prediction for sungai isap residence," *Proceedings - 2016 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2016*. pp. 236–241, 2017, doi: 10.1109/I2CACIS.2016.7885321.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[17] V. Gude, S. Corns, and S. Long, "Flood Prediction and Uncertainty Estimation Using Deep Learning," *Water (Switzerland)*, vol. 12, no. 3, 2020, doi: 10.3390/w12030884.

[18] P. Wang, J. Bai, and J. Meng, "A Hybrid Genetic Ant Colony Optimization Algorithm with an Embedded Cloud Model for Continuous Optimization," *J. Inf. Process. Syst.*, vol. 16, no. 5, pp. 1169–1182, 2020, doi: 10.3745/JIPS.01.0059.

[19] P. Goymann, D. Herrling, and A. Rausch, "Flood Prediction through Artificial Neural Networks A case study in Goslar , Lower Saxony," *Adapt. 2019 Elev. Int. Conf. Adapt. Self-Adaptive Syst. Appl.*, no. May, pp. 56–62, 2019.

[20] V. Ranganayaki and S. N. Deepa, "An intelligent ensemble neural network model for wind speed prediction in renewable energy systems," *Sci. World J.*, vol. 2016, 2016, doi: 10.1155/2016/9293529.

[21] J. Bergstra, "Algorithms for Hyper-Parameter Optimization Algorithms for Hyper-Parameter Optimization," no. May 2014, 2011.

[22] Weilisi and T. Kojima, "Investigation of Hyperparameter Setting of a Long Short-Term Memory Model Applied for Imputation of Missing Discharge Data of the Daihachiga River," *Water (Switzerland)*, vol. 14, no. 2, 2022, doi: 10.3390/w14020213.

[23] B. Yadav, S. Ch, S. Mathur, and J. Adamowski, "Assessing the suitability of extreme learning machines (ELM) for groundwater level prediction," *J. Water L. Dev.*, vol. 32, no. 1, pp. 103–112, 2017, doi: 10.1515/jwld-2017-0012.

[24] Q. F. Tan *et al.*, "An adaptive middle and long-term runoff forecast model using EEMD-ANN hybrid approach," *J. Hydrol.*, vol. 567, pp. 767–780, 2018, doi: 10.1016/j.jhydrol.2018.01.015.

[25] Z. M. Yaseen, R. C. Deo, I. Ebtehaj, and H. Bonakdari, "Hybrid Data Intelligent Models and Applications for Water Level Prediction," no. 1, pp. 121–139, 2018, doi: 10.4018/978-1-5225-4766-2.ch006.

[26] N. B. M. Khairudin, N. B. Mustapha, T. N. B. M. Aris, and M. B. Zolkepli, "Comparison of Machine Learning Models for Rainfall Forecasting," *2020 Int. Conf. Comput. Sci. Its Appl. Agric. ICOSICA 2020*, 2020, doi: 10.1109/ICOSICA49951.2020.9243275.