CAUCHY: Jurnal Matematika Murni dan Aplikasi
Volume 10(1) (2025), Pages 428-441
p-ISSN: 2086-0382; e-ISSN: 2477-3344

Combination of Extreme Learning Machine and Binary Bat Algorithm for Customer Churn Prediction

Arifin, Syaiful Anam*, Marsudi

Department of Mathematics, Faculty of Mathematics and Natural Sciences, Brawijaya University, Malang, Indonesia

Email: syaiful@ub.ac.id

ABSTRACT

One of the important assets of a company is customers. Customers determine the company's stability because they are the source of income and determine the company's competitiveness. It shows the importance of predicting which customers have the potential to switch to another company. These predictions can be made using Machine Learning (ML). One of the ML methods is the Extreme Learning Machine (ELM). The advantages of ELM compared to other methods are fast computing time, ease of use, and reach a global optimum. However, ELM has weaknesses when solving problems with high-dimensional datasets, so feature selection is required. The Binary Bat Algorithm (BBA) is a swarm intelligence method that can be used to optimize ELM performance. The advantages of BBA compared to others are few parameters and much better in effectiveness or accuracy. This research was carried out with preprocessing data, training data, and testing data. The research showed that ELM-BBA is better than ELM and ELM-Binary Particle Swarm Optimization (BPSO) in evaluation metric values. However, ELM-BBA tended to be slower than ELM-BPSO. The best results on evaluation metrics achieved by ELM-BBA were 0.97, 0.97, 0.96, and 0.97 in accuracy, precision, recall, and F1 score, respectively.

Keywords: binary bat algorithm; customer churn prediction; extreme learning machine

INTRODUCTION

One of the valuable assets of a company, including telecommunications companies, is customers. The customer is a determinant of company stability [1]. This stability is related to increasing market competitiveness and the profits obtained by the company [2]. Improvement of customer loyalty as big as 25% will impact an increase in income of up to 95% [2]. This statement is supported by research from [3] and [4] which states that the process of finding new customers is more expensive than maintaining old customers. Based on these, companies must predict whether customers will switch to another company's products or services and try to prevent it.

Customer Churn Prediction (CCP) is the process of predicting whether a customer will churn or not. Churn is a condition when a customer decides to use products or services from other companies [5]. Some factors influence customers choosing to use another company's product or service, namely poor communication between the customer and

Submitted: January 30, 2025 Reviewed: February 02, 2025 Accepted: May 08, 2025
DOI: https://doi.org/10.18860/cauchy.v10i1.31815

the company, worst response to a complaint, negative opinions about the company on social media, influence from another customer, poor service, cheaper price for products or services from competing companies, and the product is not compatible with the latest technology [6]. The sooner CCP is done, the better the performance of the company in the future [3].

CCP can be done by using Machine Learning (ML) to simplify the process. Research from [7] showed that the CCP in the fast food industry uses Support Vector Machine (SVM) with Dynamic Mutual Information (DMI) and achieves accuracy of 73.57%. Naive Bayes (NB) achieves an accuracy of 91.95% with balanced data when performing CCP on a bank [8].

One of the ML methods that can be used for CCP is Extreme Learning Machine (ELM) because it has advantages in the process. The advantages of ELM is easy to implement and can achieve global optimum [9]. Furthermore, ELM has few parameters and less computation time [10]. The following researches show that ELM is a good method for CCP. ELM-Grid Search (GS) is used to do CCP on telecommunications data and obtained an accuracy of 93.1% [10]. CCP also can be done on telecommunications data using ELM-Particle Swarm Optimization (PSO) and achieved an accuracy of 81.16% [11]. Kernelized ELM (KELM) was applied for churn predictive financial risk assessment model with Bacterial Foraging Optimization (BFO) as a hyperparameter tuning method [12]. ELM also could be used for forecasting in the insurance sector with Binary Golden Eagle Optimizer (BGEO) [13]. Research [14] used ELM and Multi-Objective Atomic Orbital Search (MOAOS) for balancing churn prediction that considers both cost and return from customers. However, there were not many types of research focused on reducing the dimensionality of the dataset so that ELM can achieve better performance. ELM works for predicting or classifying data with inverse of matrices concept so that if the dimensionality of the dataset is very large, then the process is not optimum.

ELM certainly also has weaknesses as a classification method. ELM requires large networks and computational costs when solving classification and regression problems on high-dimensional data [9]. This issue is related to the features of a dataset. If the features contained in the dataset are quite a lot, then we need a Feature Selection (FS) process. FS is done by selecting as few features as possible, but still considering the performance of model and can even improve its performance.

Various methods of swarm intelligence can improve ELM performance by doing FS on the dataset. Research [15] showed that the performance of Random Forest (RF) can be improved in cancer classification using swarm intelligence. Other research are improvement of performance Convolutional Neural Networks (CNN) using swarm intelligence for vehicle engine classification [16]. These show that swarm intelligence can be used for FS as a form of model performance optimization.

Bat Algorithm (BA) or Binary Bat Algorithm (BBA) is one of powerful swarm intelligence for solving optimization problems. The advantages of BBA compared to others are few parameters and much better in effectiveness or accuracy [17]. BA solved economic problems and concluded that BA is better than the Dragonfly Algorithm (DA) in this problem [18]. BA is better than Genetic Algorithm (GA) in robotics optimization [19]. Research [20] said that BA outperforms the Firefly Algorithm (FA), Cuckoo Search Algorithm (CSA), and Harmony Search Algorithm (HSA) in the FS process. BBA is used for FS thereby increasing SVM accuracy up to 99.28% [21]. BBA is used for FS and got conclusion that BBA can improve several classifiers well [17]. Several studies in the previous became the inspiration for selecting BBA for FS so that BBA can improve the performance of ELM in CCP.

BBA in this paper is a feature selection method in the case of customer churn prediction using ELM. It is important research because commonly data for predicting customer churn have many features making the performance of ELM does not achieve optimum results. The ELM-BBA method proposed in this study will try two activation functions in ELM, namely sigmoid and tanh. In addition, this research tries several conditions with different numbers of nodes in ELM, namely 20 nodes, 40 nodes, and 80 nodes. The proposed BBA also applies a higher exploration phase by utilizing a random number as a threshold to convert the value into binary. This certainly makes the computational process longer, but the results achieved are maximized. As a solution so that the model can still work with less time, the model will add convergence criteria with certain tolerance and counter values so that the model will stop if it is considered convergent even though it has not reached the maximum iteration.

ELM combined with BBA, hopefully, can be an innovation that can help the company to predict whether their customer switch to another company or not. Certainly, the proposed model has excellent performance so that the company can make good decisions based on CCP with the proposed model.

The summary of this research contributions are as follows:
1. Proposing a new model for CCP using ELM and BBA. BBA is used as a FS method such that the dimensions of the dataset are reduced.
2. Reducing overfitting of ELM for CCP with BBA.
3. Enhancing the accuracy, precision, recall, and F1 score of ELM for CCP with BBA.
4. Proposing a new ELM model that consistently achieves high performance for CCP, even with a few nodes, and is not limited to a single type of activation function.

METHODS

This research uses a dataset from an open source website, namely Kaggle. The dataset is downloaded in .csv format and has 7043 records and 52 features. The list of these features can be seen in Table 1. The target feature of the dataset is Churn. The value of the target feature is a number 0 or 1. If the value of the feature is 0, then the customer will not switch to another company's products or services. If the value of the feature is 1, then the customer switch to another company's products or services. The dataset used in this research is imbalanced because there are 73.46% non-churn records and 26.54% churn records. There are several features related to the condition of whether a customer churns or not, namely Churn Category, Churn Reason, Churn Score, and Churn. The Churn Category describes the customer's reason for moving to another company briefly. Churn Reason is a feature that explains the customer's reason for moving to another company in detail. Churn Score is a feature containing a value of $0 - 100$ which states the chances of customers moving to another company. The greater the value, the greater the chance of customers moving to another company.

Table 1. Features of Telco Customer Churn Dataset

No	Feature	Data Type	No	Feature	Data Type
1	Age	Numerical	27	Online Backup	Numerical
2	Avg Monthly GB Download	Numerical	28	Online Security	Categorical
3	Avg Monthly Long Distance	Numerical	29	Paperless Billing	Categorical
4	Churn Category	Categorical	30	Partner	Categorical

No	Feature	Data Type	No	Feature	Data Type
5	Churn Reason	Categorical	31	Payment Method	Categorical
6	Churn Score	Categorical	32	Phone Service	Categorical
7	City	Numerical	33	Population	Categorical
8	CLTV	Categorical	34	Premium Tech Support	Categorical
9	Contract	Categorical	35	Quarter	Numerical
10	Country	Numerical	36	Referred a Friend	Categorical
11	Customer ID	Categorical	37	Satisfaction Score	Categorical
12	Customer Status	Categorical	38	Senior Citizen	Categorical
13	Dependents	Categorical	39	State	Numerical
14	Device Protection Plan	Categorical	40	Streaming Movies	Categorical
15	Gender	Categorical	41	Streaming Music	Categorical
16	Internet Service	Categorical	42	Streaming TV	Categorical
17	Internet Type	Categorical	43	Tenure in Months	Categorical
18	Lat Long	Categorical	44	Total Charges	Categorical
19	Latitude	Categorical	45	Total Extra Data Charges	Numerical
20	Longitude	Numerical	46	Total Long Distance Charges	Numerical
21	Married	Numerical	47	Total Refunds	Numerical
22	Monthly Charge	Numerical	48	Total Revenue	Numerical
23	Multiple Lines	Categorical	49	Under 30	Numerical
24	Number of Dependent	Numerical	50	Unlimited Data	Numerical
25	Number of Referral	Categorical	51	Zip Code	Categorical
26	Offer	Numerical	52	Churn	Categorical

The stage of this research consists of several steps, namely data preprocessing, data splitting, training ELM-BBA, training comparing model, testing ELM-BBA, testing ELM-BBA, and comparing the ELM-BBA model with comparing model (ELM without FS and ELM-BPSO). ELM-BPSO is a method that combines ELM as a classification method and Binary Particle Swarm Optimization (BPSO) as an optimization method for feature selection. This method will be a comparison method for the proposed method. The research stage can be seen in Figure 1.

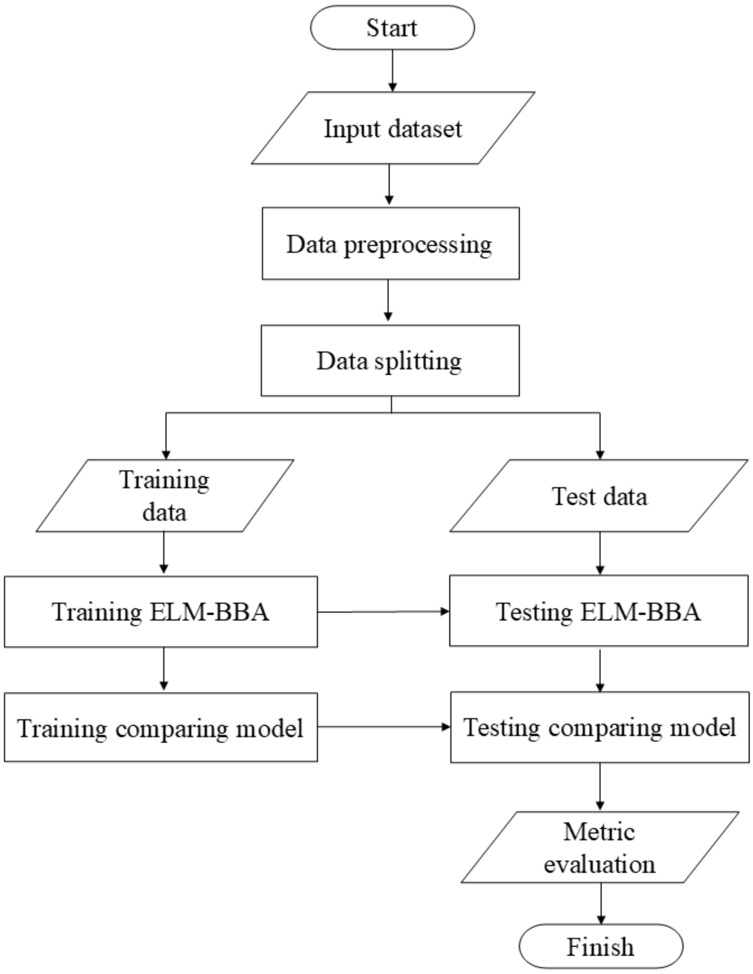


Figure 1. Research stage

Data Preprocessing

This process is carried out in several stages. The first is handling missing values. If the number of missing items is less than 30%, then the missing values can be filled by mode or median [22]. Otherwise, features or records with more than or equal to 30% missing values can be dropped [23]. The second is decoding categorical features. The third is normalization with Min-Max Normalization. The last is the detection of outliers using the Inter Quartile Range (IQR) method.

Data Splitting

Data distribution is done in proportion to 80% for training data and 20% for testing data. After that, the process is continued by data balancing using Synthetic Minority Oversampling Technique-Nominal Continuous (SMOTE-NC) on training data so that the model can recognize the minority class as well as the majority class [24]. This process also ensures that the proportion of non-churn record and churn record have same proportion on training data and testing data.

Bat Algorithm

Bat Algorithm (BA) is an algorithm inspired by the movement patterns of bats. BA was proposed by Yang in 2009 [25]. Bats use echolocation techniques to find food and avoid predators. Yang developed BA based on the characteristics of bats [26]. These characteristics are as follows.

1. Bats can determine the distance between their body position and objects in front of them and distinguish between objects and prey.
2. Bats can fly randomly with velocity $\boldsymbol{v}$ at position $\boldsymbol{x}$, frequency f, pulse rate r, and loudness A for finding prey. Pulse rate and loudness may vary based on the prey's location.
3. Loudness varies from A_0 (large positive value) to A_{min} (constant minimum value)

The velocity of the bat i on the t -th iteration ($\boldsymbol{v}_i^t$) and position of the bat i on the t -th iteration ($\boldsymbol{x}_i^t$) are updated with formulas like the following.

$$f_i = f_{min} + (f_{max} - f_{min})\beta, \tag{1}$$

$$\boldsymbol{v}_i^t = \boldsymbol{v}_i^{t-1} + (\boldsymbol{x}_i^t - \boldsymbol{x}_*)f_i, \tag{2}$$

$$\boldsymbol{x}_i^t = \boldsymbol{x}_i^{t-1} + \boldsymbol{v}_i^t, \tag{3}$$

β is random number from distribusi $U[0,1]$, and $\boldsymbol{x}_*$ is best solution. The frequency may vary from f_{max} to f_{min} randomly. The loudness of bat i (A_i) and the pulse rate of bat i (r_i) are used for controlling exploration and exploitation mechanisms. The closer the bat to its prey, the smaller the loudness and the bigger the pulse rate. A_i and r_i are updated with formulas like the following.

$$A_i^{t+1} = \alpha A_i^t, \tag{4}$$

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)], \tag{5}$$

$\alpha \in [0,1]$ and $\gamma > 0$ are constant parameters.

Binary Bat Algorithm

BA does not require the best position with real number value in some case. BA requires the best position with binary value (0 or 1) as in the FS cases. This condition shows that BA needs to modify the position of the bat in BA into a vector containing binary numbers [27].

The process of converting real values to binary can be done in several ways. According to [27], transforming the bat position values to binary values can be done using the sigmoid function found in Equation 6.

$$S(v_{ij}^t) = \frac{1}{1 + e^{-v_{ij}^t}}. \tag{6}$$

Next, the formula for updating the bat position on Equation (3) will be changed with the formula like the following.

$$x_{ij}^t = \begin{cases} 1 & , \; S(v_{ij}^t) > \sigma \\ 0 & , \quad lainnya \end{cases}, \tag{7}$$

S is the sigmoid function, v_{ij}^t is the velocity of the bat i on j -th dimension and t -th iteration, x_{ij}^t is the position of the bat i on j -th dimension and t -th iteration, e is the Euler number, and σ is the random number from $U[0,1]$.

Extreme Learning Machine

One reliable method for training artificial feedforward neural networks with a single hidden layer is an Extreme Learning Machine (ELM). The initialization of hidden nodes for ELM is done randomly and the important essence of ELM is that the adjustment of the hidden layer does not require iteration [9]. This makes ELM reduce the time for training, easy to use, and able to find the global optimum well. However, ELM requires more hidden nodes than conventional neural networks [28].

The algorithm of ELM is based on the training problem of a single hidden layer feedforward neural network [29]. The detailed algorithm is as follows.

1. For N arbitrary different samples $(\boldsymbol{x_i}, \boldsymbol{t_i})$ with $(\boldsymbol{x_i}, \boldsymbol{t_i}) \in \mathbb{R}^{\boldsymbol{n}} \times \mathbb{R}^{\boldsymbol{m}} (i = 1,2, \dots, N)$, *single hidden layer feedforward neural networks* (SLFNs) with $\widetilde{N}$ hidden nodes and activation function $f(x)$ is modeled mathematically as

$$\sum_{i=1}^{\widetilde{N}} \boldsymbol{\beta_i} f_i(\boldsymbol{x_j}) = \sum_{i=1}^{\widetilde{N}} \boldsymbol{\beta_i} f_i(\boldsymbol{a_i} \cdot \boldsymbol{x_j} + b_i) = \boldsymbol{t_j}, j = 1, \dots, N, \tag{8}$$

   with $\boldsymbol{a_i} = [a_{i1}, a_{i2}, \dots, a_{in}]^T$ is the weight vector connected with i -th hidden node and input nodes, and b_i is bias from i -th hidden node. $\boldsymbol{\beta_i} = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ is the weight vector connected i -th hidden node and output nodes. $\boldsymbol{a_i} \cdot \boldsymbol{x_j}$ represents the inner product of $\boldsymbol{a_i}$ and $\boldsymbol{x_i}$, and the activation function commonly uses sigmoid, sinus, tanh, or other functions.
2. Changing of Equation (8) can be simplified so that

$$H\boldsymbol{\beta} = \boldsymbol{T}, \tag{9}$$

   with $H(\boldsymbol{a_1}, \dots, \boldsymbol{a_{\widetilde{N}}}, b_1, \dots, b_{\widetilde{N}}, \boldsymbol{x_1}, \dots, \boldsymbol{x_N}) = \begin{pmatrix} f(\boldsymbol{a_1} \cdot \boldsymbol{x_1} + b_1) & \cdots & f(\boldsymbol{a_{\widetilde{N}}} \cdot \boldsymbol{x_1} + b_{\widetilde{N}}) \\ \vdots & \cdots & \vdots \\ f(\boldsymbol{a_1} \cdot \boldsymbol{x_N} + b_1) & \cdots & f(\boldsymbol{a_{\widetilde{N}}} \cdot \boldsymbol{x_N} + b_{\widetilde{N}}) \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_1^T \\ \vdots \\ \beta_{\widetilde{N}}^T \end{pmatrix}_{\widetilde{N} \times m}, \boldsymbol{T} = \begin{pmatrix} t_1^T \\ \vdots \\ t_N^T \end{pmatrix}_{N \times m}.$
3. Finding $\boldsymbol{\beta}$ value from Equation (9) using Moore-Penrose inverse like equation following.

$$\boldsymbol{\beta} = H^{\dagger}\boldsymbol{T}, \tag{10}$$

   with $H^{\dagger}$ represent generalized Moore-Penrose inverse from matrix H.

$\boldsymbol{w_i}, b_i$, and $\boldsymbol{\beta_i}$ is used for testing test data that has been provided.

Metric Evaluation

The evaluation metrics that will be used in this study are: accuracy, precision, recall, and F1 score. The values of the four-evaluation metrics are between 0 and 1. If the value gets closer to 1, then the model shows better performance. According to [30], the definition of each type of evaluation metric is as follows.

1) Accuracy is a metric that measures how many people are correctly classified out of the total number of people. The formula for accuracy is

$$accuracy = \frac{TN + TP}{TN + TP + FP + FN}. \tag{11}$$

2) Precision is the ratio of people correctly classified as positive to the total number of people classified as positive. The formula for precision is

$$precision = \frac{TP}{TP + FP}. \tag{12}$$

3) Recall is the ratio of people who are classified as positive to the total number of people who have the disease. The formula for the recall is

$$recall = \frac{TP}{TP + FN}. \tag{13}$$

4) F1 score is the harmonic mean of the precision and recall which can state the accuracy of the method in each class. The formula for the F1 score is

$$F1\ score = 2 \times \frac{recall \times precision}{recall + precision}. \tag{14}$$

Another metrics that can be used in model evaluation is computation time. The less computation time required, the better the proposed model is said to be.

RESULTS AND DISCUSSION

Data preprocessing and data splitting change the number of data records. After these processes, there were 8278 training data and 1409 testing data. In this research, there were several combinations of ELM hyperparameters (number of nodes and activation functions). The number of nodes that will be tried were $20, 40,$ and $80,$ while the activation functions that will be studied were sigmoid and tanh. The parameter of BBA was set to refer to Yang's research (2019) [25]. Parameters of the BBA used in this research can be seen in Table 2.

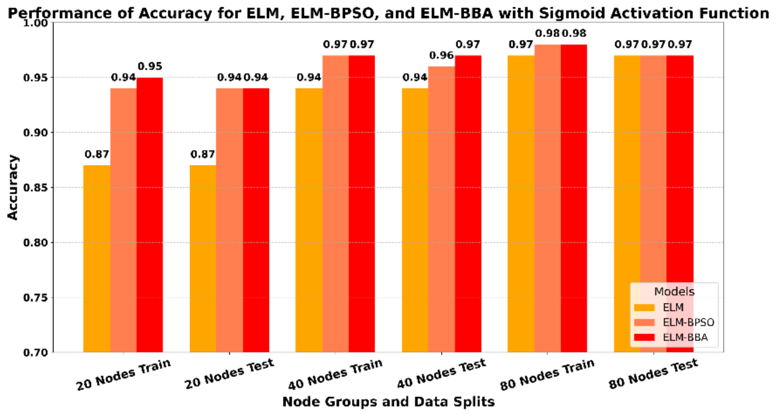
The first experiment in this research is CCP using ELM without FS. This experiment was carried out six times with different conditions based on the combination of ELM hyperparameters. The second and third experiments also was carried out six times with different conditions, but FS was done for optimizing ELM using BBA in the second experiment and BPSO in the third experiment. Each experiment was done 25 times and each iteration was carried out cross-validation to test the performance of features from both training and testing data using 5-fold. All of the experiment result is displayed based on average metric evaluation and will be compared with each other.

Table 2. Parameter of BBA

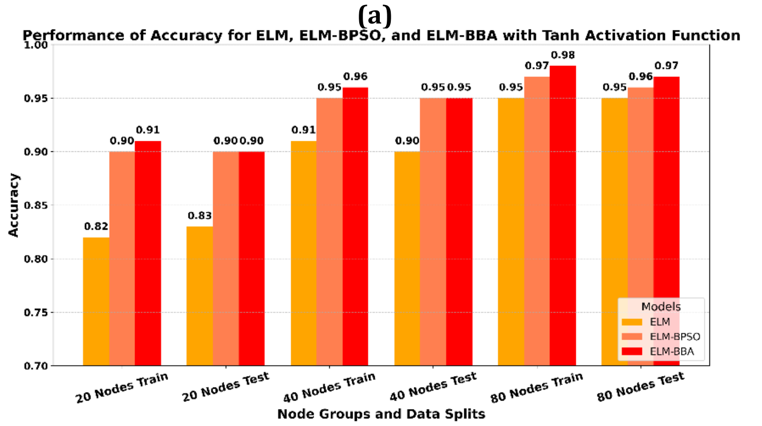
Parameter	Value
Number of bats	100
Maximum iteration	50
Initialization of loudness (A_0)	0.95
Initialization of pulse rate (r_0)	0.9
Decreasing loudness coefficient (α)	0.99
Increasing pulse rate coefficient (γ)	0.9
Minimum frequency	0
Maximum frequency	5
Dimensions	52
Lower bound	$[0, 0, \dots, 0]^{52}$
Upper bound	$[1, 1 \dots, 1]^{52}$

The accuracy performance of the training and the testing process can be seen in Figure 2. This figure showed that the BPSO and BBA were able to improve ELM performance in both the training and testing processes. This is increasingly visible when ELM uses 20 or 40 nodes. The use of the activation function also affected improving ELM performance. The use of sigmoid as an activation function had shown good results before being optimized. This was different from the performance of ELM when using tanh as the activation function. BPSO and BBA seemed to have a bigger impact in improving ELM performance so that accuracy after optimization can be equivalent to ELM with sigmoid as the activation function. However, the value of the training accuracy of ELM with 20 nodes and the testing accuracy of ELM with 40 nodes were higher when ELM was combined with BBA when ELM was combined with BPSO.

The precision performance of the training and the testing process can be seen in Figure 3. The trend in average precision values from the ELM, ELM-BPSO, and ELM-BBA models also showed similar results to the accuracy values. ELM-BPSO and ELM-BBA appeared to outperform ELM when ELM uses 20 or 40 nodes and tanh activation function. If ELM used 80 nodes and a sigmoid activation function, it only increased by 0.01 after optimization with both BPSO and BBA in the precision of the testing process. However, specifically, the precision value was similar to the accuracy value in both models. This means that the model was able to categorize customers churn correctly compared with all customer that categorized churn by model.

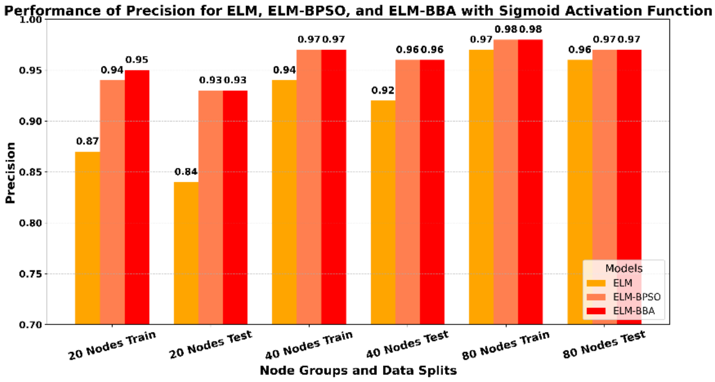


(a)

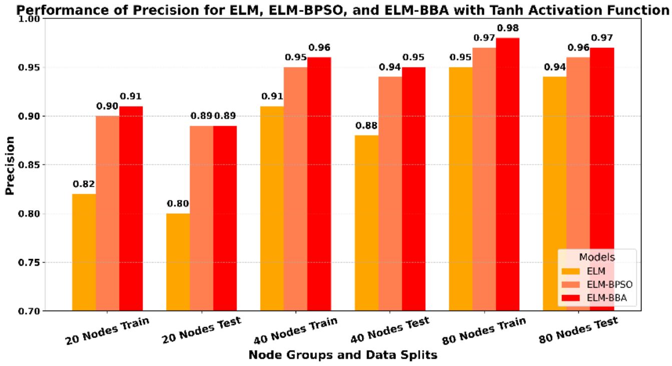


(b)

Figure 2. Accuracy performance of training and testing data with different activation functions: (a) sigmoid (b) tanh



(a)



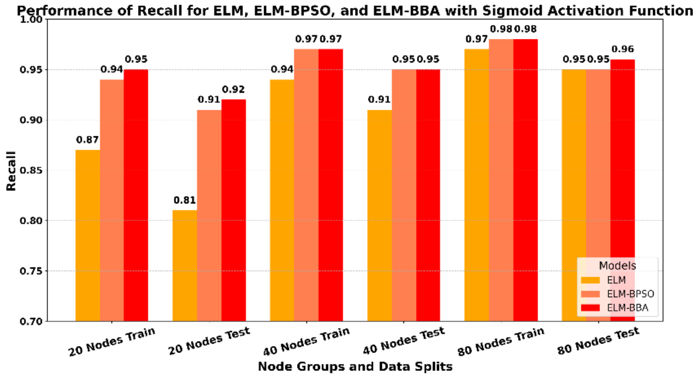
(b)

Figure 3. Precision performance of training and testing data with different activation functions: (a) sigmoid (b) tanh

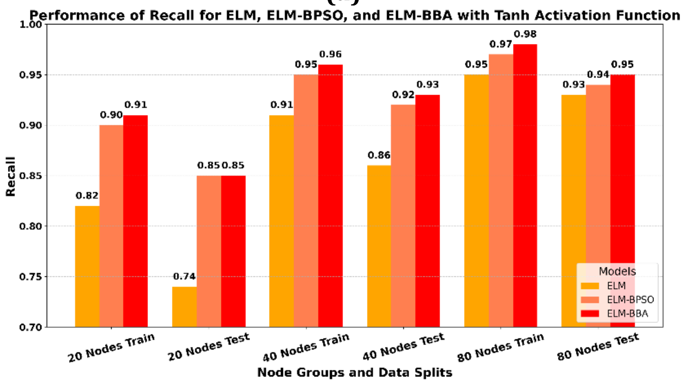
The third evaluation metric was an average recall. Recall the performance of the training and the testing process can be seen in Figure 4. The same as the accuracy metric and precision metric, the recall metric showed the superiority of ELM with 80 nodes over ELM with 20 or 40 nodes because ELM with 80 nodes had shown good performance, even though without optimization. However, the more nodes the model, the more complex it will be and require high computing and storage time. Apart from that, it was also apparent that the sigmoid activation function is superior to the tanh activation function. The value of the recall metric of ELM-BPSO and ELM-BBA were generally less than the value of the accuracy metric and precision metric of these models, but the value of the recall metric was high compared with ELM without optimization. This shows that the model is capable enough to recognize customers who have the potential to churn. ELM-BBA had higher recall values than ELM-BPSO on most evaluation metrics, especially when ELM used the tanh activation function.

The final evaluation metric was the average the F1 score which strengthens the results presented from the three previous evaluation metrics. F1 score performance of the training and the testing process can be seen in Figure 5. The testing values of F1 score before and after optimization using BPSO or BBA were still below the training values for all ELM models with six hyperparameter combinations. This shows that there was overfitting due to the characteristics of the dataset used and the complexity of the model. However, it can be seen that BBA and BPSO combined with ELM were able to reduce the level of overfitting. This showed that BBA plays a role in improving the performance of the ELM model.

In general, ELM-BBA outperformed ELM-BPSO in all evaluation metrics, but ELM-BBA's superiority was less visible when ELM used 80 nodes. Besides that, ELM-BBA's superiority was less visible when ELM used the sigmoid activation function. Furthermore, ELM-BBA was able to increase all evaluation metric values and does not depend on the number of nodes or activation functions used compared to ELM without optimization and ELM with BPSO.

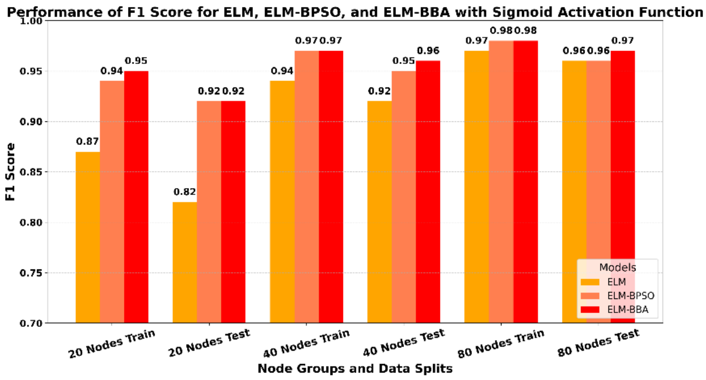


(a)

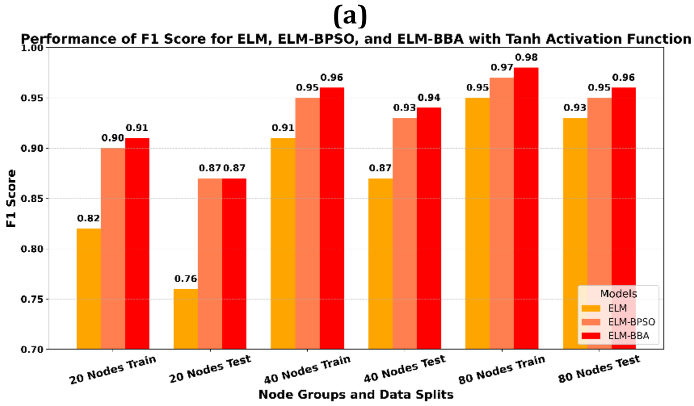


(b)

Figure 4. Recall the performance of training and testing data with different activation functions: (a) sigmoid (b) tanh



(a)



(b)

Figure 5. F1 score performance of training and testing data with different activation functions: (a) sigmoid (b) tanh

ELM-BBA which outperformed ELM-BPSO on evaluation metrics was not superior in average computing time for the training model. ELM-BPSO turns out to be faster than ELM-BBA. This was possibly caused by ELM-BPSO which tended to be quicker in determining solutions that were not necessarily the best so ELM-BPSO was fast in the FS process, but did not produce maximum output. In addition, BBA has more parameters and more processes to determine the best solution than BPSO so BBA requires longer computation time than BPSO [26]. The computing times for the ELM-BPSO and ELM-BBA training process can be seen in Table 3. It can be seen that the more nodes used in ELM, the longer the time consumption required. On the other sides, using tanh also required more time than using sigmoid. In this research, there was a time anomaly when ELM used 40 nodes and the sigmoid. When optimized with BPSO, the time required was slower than ELM used 80 nodes. When optimized with BBA, the time required was faster than ELM used 20 nodes. This might happen because there was a random process of generating values and there were convergence criteria that allowed the process to stop even though it had not reached the maximum iteration.

Table 3. Summary of ELM-BPSO and ELM-BBA Time Computation

Hyperparameter	Time of BPSO (s)		Time of BBA (s)	
	Average	Standard Deviation	Average	Standard Deviation
[20, Sigmoid]	43.14	10.93	205.59	71.64
[20, Tanh]	71.33	26.97	211.66	66.75
[40, Sigmoid]	81.53	23.86	180.02	68.06
[40, Tanh]	209.07	237.65	574.88	171.91
[80, Sigmoid]	71.15	43.96	573.23	158.76
[80, Tanh]	357.54	150.65	1003.72	362.15

CONCLUSIONS

ELM is an excellent machine learning method for classification, including CCP. ELM with the right nodes and activation functions can classify problems optimally. BBA as a swarm intelligence method that can be used for FS can be combined with ELM to become the proposed model, namely ELM-BBA to provide high performance on ELM even though it uses few nodes and is not limited to just one activation function. ELM-BBA shows better results than ELM-BPSO from all evaluation metric values, although in terms of computing time ELM-BBA is slower than BPSO. The best results on evaluation metrics achieved by BBA were an average accuracy score of 0.97, average precision of 0.97, average recall of 0.96, average F1 score of 0.97 with 573.23 seconds average computation time. Several limitations need to be addressed in future research. Firstly, the computational time of ELM-BBA is significantly longer compared to ELM-BPSO. Future studies could explore advanced optimization techniques or parallel computing frameworks to reduce the computational burden while maintaining high accuracy. Secondly, the scalability of ELM-BBA on larger and more complex datasets remains unexplored. Testing the model on high-dimensional data or datasets would provide insights into its robustness and applicability in real-world applications.

REFERENCES

[1] I. Ranggadara, G. Wang, and E. R. Kaburuan, "Applying Customer Loyalty Classification with RFM and Naïve Bayes for Better Decision Making," *Proc. - 2019 Int. Semin. Appl. Technol. Inf. Commun. Ind. 4.0 Retrosp. Prospect. Challenges, iSemantic 2019*, pp. 564–568, 2019, doi: 10.1109/ISEMANTIC.2019.8884262.

[2] X. Xiahou and Y. Harada, "B2C E-Commerce Customer Churn Prediction Based on," pp. 458–475, 2022.

[3] A. K. Ahmad, A. Jafar, and K. Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0191-6.

[4] N. Hazimah, S. Harahap, A. Amirullah, M. B. Saputro, and I. A. Tamaroh, "Classification of potential customers using C4.5 and k-means algorithms to determine customer service priorities to maintain loyalty," *J. Soft Comput. Explor.*, vol. 3, no. 2, pp. 123–130, 2022, doi: 10.52465/joscex.v3i2.89.

[5] M. Imani, "Customer Churn Prediction in Telecommunication Industry: A Literature Review," 2024, doi: 10.20944/preprints202403.0585.v1.

[6] Wael Fujo Samah, Subramanian Suresh, and Ahmad Khder Moaiad, "Customer Churn Prediction in Telecommunication Industry Using Deep Learning," *Inf. Sci. Lett.*, vol. 11, no. 1, pp. 1–15, 2022.

[7] H. Sulistiani, K. Muludi, and A. Syarif, "Implementation of Dynamic Mutual Information and Support Vector Machine for Customer Loyalty Classification," *J. Phys. Conf. Ser.*, vol. 1338, no. 1, 2019, doi: 10.1088/1742-6596/1338/1/012050.

[8] V. Agarwal, S. Taware, S. A. Yadav, D. Gangodkar, A. L. N. Rao, and V. K. Srivastav, "Customer - Churn Prediction Using Machine Learning," *Proc. Int. Conf. Technol. Adv. Comput. Sci. ICTACS 2022*, pp. 893–899, 2022, doi: 10.1109/ICTACS56270.2022.9988187.

[9] R. Kaur, R. K. Roul, and S. Batra, "Multilayer extreme learning machine: a systematic review," *Multimed. Tools Appl.*, vol. 82, no. 26, pp. 40269–40307, 2023, doi: 10.1007/s11042-023-14634-4.

[10] F. Ö. Koçoğlu and T. Özcan, "A grid search optimized extreme learning machine approach for customer churn prediction," *J. Eng. Res.*, vol. 11, no. 3, pp. 103–112, 2022, doi: 10.36909/jer.16771.

[11] K. G. Li and B. P. Marikannan, "Hybrid particle swarm optimization-extreme learning machine algorithm for customer churn prediction," *J. Comput. Theor. Nanosci.*, vol. 16, no. 8, pp. 3432–3436, 2019, doi: 10.1166/jctn.2019.8304.

[12] U. M. F. Dimlo, R. Huerta-Soto, L. Nivin-Vargas, J. Tarazona-Jiménez, C. Reyes-Reyes, and N. Girdharwal, "Kernelized Extreme Learning Machine Enabled Churn Predictive Financial Risk Assessment Model," *Proc. - Int. Conf. Augment. Intell. Sustain. Syst. ICAISS 2022*, pp. 412–417, 2022, doi: 10.1109/ICAISS55157.2022.10010834.

[13] N. Jajam and N. P. Challa, "Dynamic Behavior-Based Churn Forecasts in the Insurance Sector," *Comput. Mater. Contin.*, vol. 75, no. 1, pp. 977–997, 2023, doi: 10.32604/cmc.2023.036098.

[14] P. Jiang, Z. Liu, L. Zhang, and J. Wang, "Hybrid model for profit-driven churn prediction based on cost minimization and return maximization," *Expert Syst. Appl.*, vol. 228, no. January 2022, p. 120354, 2023, doi: 10.1016/j.eswa.2023.120354.

[15] S. Jeyasingh and M. Veluchamy, "Modified bat algorithm for feature selection with the Wisconsin Diagnosis Breast Cancer (WDBC) dataset," *Asian Pacific J. Cancer Prev.*, vol. 18, no. 5, pp. 1257–1264, 2017, doi: 10.22034/APJCP.2017.18.5.1257.

[16] Y. Li, Y. Zhao, Y. Shang, and J. Liu, "An improved firefly algorithm with dynamic self-adaptive adjustment," *PLoS One*, vol. 16, no. 10 October 2021, pp. 1–24, 2021, doi: 10.1371/journal.pone.0255951.

[17] F. Liu, X. Yan, and Y. Lu, "Feature Selection for Image Steganalysis Using Binary Bat Algorithm," *IEEE Access*, vol. 8, pp. 4244–4249, 2020, doi: 10.1109/ACCESS.2019.2963084.

[18] K. Sumanth and M. V. Priya, "Finding Solution for Practical Economic Load Dispatch Problem Using Dragonfly Algorithm in Comparison with Novel Bat Algorithm," *Proc. Int. Conf. Artif. Intell. Knowl. Discov. Concurr. Eng. ICECONF 2023*, pp. 1–4, 2023, doi: 10.1109/ICECONF57129.2023.10083769.

[19] C. A. Griffiths, C. Giannetti, K. T. Andrzejewski, and A. Morgan, "Comparison of a Bat and Genetic Algorithm Generated Sequence against Lead through Programming When Assembling a PCB Using a Six-Axis Robot with Multiple Motions and Speeds," *IEEE Trans. Ind. Informatics*, vol. 18, no. 2, pp. 1102–1110, 2022, doi: 10.1109/TII.2021.3082877.

[20] V. Yassaswini and S. Baskaran, "An Optimization of Feature Selection for Classification using Modified Bat Algorithm," *Int. J. Inf. Technol. Comput. Sci.*, vol. 13, no. 4, pp. 38–46, 2021, doi: 10.5815/ijitcs.2021.04.04.

[21] R. Yaghoubzadeh, S. Kamel, H. Barzegar, and B. San'ati, "The Use of the Binary Bat Algorithm in Improving the Accuracy of Breast Cancer Diagnosis," *Multidiscip. Cancer Investig.*, vol. 5, no. 1, pp. 1–8, 2021, doi: 10.30699/mci.5.1.372-2.

[22] M. Karanovic, M. Popovac, S. Sladojevic, M. Arsenovic, and D. Stefanovic, "Telecommunication Services Churn Prediction - Deep Learning Approach," *2018 26th Telecommun. Forum, TELFOR 2018 - Proc.*, no. January 2019, 2018, doi: 10.1109/TELFOR.2018.8612067.

[23] U. Sa'adah, M. Y. Rochayani, D. W. Lestari, and D. A. Lusia, *Kupas Tuntas Algoritma Data Mining dan Implementasinya Menggunakan R*. Malang: Universitas Brawijaya Press, 2021.

[24] D. T. Utari, "Integration of Svm and Smote-Nc for Classification of Heart Failure Patients," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 17, no. 4, pp. 2263–2272, 2023, doi: 10.30598/barekengvol17iss4pp2263-2272.

[25] Y. Wang *et al.*, "A novel bat algorithm with multiple strategies coupling for numerical optimization," *Mathematics*, vol. 7, no. 2, pp. 1–17, 2019, doi: 10.3390/math7020135.

[26] T. Agarwal and V. Kumar, "A Systematic Review on Bat Algorithm: Theoretical Foundation, Variants, and Applications," *Arch. Comput. Methods Eng.*, vol. 29, no. 5, pp. 2707–2736, 2022, doi: 10.1007/s11831-021-09673-9.

[27] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, and J. P. Papa, "BBA : A Binary Bat Algorithm for Feature Selection," *2012 25th SIBGRAPI Conf. Graph. Patterns Images*, pp. 291–297, 2012, doi: 10.1109/SIBGRAPI.2012.47.

[28] B. Deng, X. Zhang, W. Gong, and D. Shang, "An overview of extreme learning machine," *Proc. - 2019 4th Int. Conf. Control. Robot. Cybern. CRC 2019*, pp. 189–195, 2019, doi: 10.1109/CRC.2019.00046.

[29] J. Wang, S. Lu, S. H. Wang, and Y. D. Zhang, "A review on extreme learning machine," *Multimed. Tools Appl.*, vol. 81, no. 29, pp. 41611–41660, 2022, doi: 10.1007/s11042-021-11007-7.

[30] M. Nasser and U. K. Yusof, "Deep Learning Based Methods for Breast Cancer Diagnosis: A Systematic Review and Future Direction," *Diagnostics*, vol. 13, no. 1, 2023, doi: 10.3390/diagnostics13010161.