Hybrid of AdaBoost and Optimized LSTM Using Modified Bald Eagle Search in Predicting Concrete Compressive Strength

Herny Christine[1] and Rila Mandala[2]

[1]Faculty of Computing, President University, Cikarang, Indonesia
[2]School of Electrical Engineering and Informatics, U-CoE AI-VLB,
Institut Teknologi Bandung, Bandung, Indonesia
herny.christine@student.president.ac.id, rila@itb.ac.id

Abstract: Nowadays, many machine learning techniques are widely integrated into human life sectors such as transportation, industries, and others. LSTM has been vastly used to forecast sequential problems. Hybridization between several models has started to emerge recently to improve the model performance due to its capabilities compared to single model. Latest researches have applied hybridization of LSTM model with other algorithms such as LSTM-AdaBoost, LSTM-BES, and others. However, the optimization and exploration in selecting the learning parameters of the model can be subsequently improved to increase the accuracy. Therefore, this paper aims to propose a hybrid model of AdaBoost and optimized LSTM using modified Bald Eagle Search algorithm to predict concrete compressive strength as the dataset. Modified Bald Eagle Search algorithm is incorporated to find the optimum solution and parameters of LSTM model. By combining the behaviour of AdaBoost into the model and using Quasi opposition-based learning to enhance the performance of Bald Eagle Search algorithm, thus, it can further improve the accuracy in predicting concrete compressive strength. The research results have shown that the proposed model has obtained better MSE, MAE, and MAPE compared to LSTM, LSTM-BES, and LSTM-MBES model.

Keywords: AdaBoost; LSTM; modified bald eagle search; quasi opposition-based learning; concrete compressive strength

1. Introduction

Nowadays, many machine learning techniques are widely integrated into human life sectors such as transportation, industries, and others. The approach of performing hybridization between several models has started to emerge to improve model performance and combine capabilities from each model. One of the models is LSTM which has been widely used to forecast sequential problems such as weather prediction, stock market prediction, and others.
The learning capability of LSTM to handle vanishing gradient problems has provided several impacts which lead to different fields of applications using LSTM [1]. Research in financial market predictions have shown that LSTM outperforms benchmark model which is Random Forest [2]. In forecasting crude oil price, LSTM is used and results have shown that LSTM is superior compared to other models as it can analyze the fluctuation pattern of crude oil price [3]. Some related works have shown that performing hybridization of LSTM manages to improve the model performance compared to a single LSTM model. A study by Li, X., Zhang, L., Wang, Z., and Dong, P [4] proposed a novel Elman-LSTM to forecast remaining life of lithium batteries. The experiments have shown that the combination of both methods results in better performance of the model compared to other state-of-art models. In addition, Wu, Y. X., Wu, Q. B., and Zhu, J. Q [5] proposed LSTM and ensemble empirical model decomposition to improve the prediction of crude oil price movement. The proposed model is proven to be effective as it obtained the lowest RMSE and MAPE.

Several related studies performed hybridization and optimization of LSTM with various methods such as AdaBoost. A study [6], predicts sea surface temperature through combining AdaBoost and LSTM method. Prediction results from both models are averaged as the final predictions. Rolling prediction scheme is also implemented in the study to find patterns in sea surface temperature. Overall, it able to improve the RMSE values of the predictions, however,

Received: January 20th, 2023. Accepted: December 2nd, 2023
DOI: 10.15676/ijeei.2023.15.4.2

in certain locations the RMSE starts to increase due to variations in the data and rolling prediction scheme. Another study [7], proposed a hybrid ensemble learning method for forecasting financial time series using AdaBoost and LSTM. It outperforms the single models and produces 19.44% - 22.33% better directional forecasts than single models. It also significantly improves the performance and reaches the accuracy 76.88% for exchange rates forecasting.

Recently, metaheuristic algorithms have started to gain a lot of popularity due to their abilities in searching for optimum solution. However, single metaheuristics algorithms might experience slow convergence rate and earlier prone to convergence situation, thus there is a need to perform hybridization [8]. Some researches optimized LSTM through the metaheuristic algorithm such as Bald Eagle Search algorithm. In the study [9], it modifies the Bald Eagle Search algorithm to find the optimized LSTM hyperparameters to predict wind energy. Empirical Model Decomposition (EMD) is also used to pre-process the data and improves the forecasting results. It produces a significant result in terms of accuracy rate compared to LSTM, however, some external factors such as atmosphere pressure will affect the prediction result. Another study [10], proposed VMD-IBES-LSTM model to predict the price of aquatic products. Variational Model Decomposition (VMD) was chosen to decompose data and overcome non-stationary series. Levy flight and tent mapping strategy were used to improve the Bald Eagle Search algorithm when initializing population and performing local search. The proposed model has a better performance and lower error compared to other individuals in the experiments, however, some errors in accuracy happen due to fluctuations in the price.

This study is using concrete compressive strength dataset [11] to evaluate the model performance, there are some related studies which predict the concrete compressive strength using various machine learning methods. A study [12] proposed an optimized evolutionary artificial neural network algorithm; a combination of ANN and genetic algorithm to predict concrete compressive strength. The result has shown that it achieved correlation coefficient R2 in training, testing, and validation which are 0.910, 0.935, and 0.899. It has higher accuracy and flexibility compared to multiple regression model. In study [13], using LSTM model to predict concrete compressive strength for High Strength Concrete (HSC), able to capture the non-linearity relationship between components. In addition, it achieved R2 score of 0.99 and compared with SVR, the prediction made by LSTM is more accurate. Another study [14], compared the feasibility between adaptive neuro-fuzzy inference system (ANFIS), artificial neural network (ANN), and multiple linear regression to predict 28 days concrete compressive strength. Both ANFIS and ANN achieved higher R2 score compared to multiple linear regression model. Also, it is not feasible enough for multiple linear regression model to predict, due to the non-linearity relationship between the components.

Another study [26] proposed an improved EM algorithm and ridge regression method, which achieved higher accuracy and maximum error 0.37 mpa when predicting concrete compressive strength compared to the original algorithm. In study [27], it proposed using tree-based methods in predicting concrete compressive strength, has shown better R2 score compared to artificial neural network (ANN) method. Overall, there are many researches on hybrid approaches related to predict concrete compressive strength. However, especially in the case of LSTM model when predict concrete compressive strength, the optimization and selection of hyperparameters could be subsequently improved to provide better optimum solution. Therefore, this paper aims to propose a hybrid model of AdaBoost and optimized LSTM using modified Bald Eagle Search algorithm to predict concrete compressive strength as the dataset.

The main contributions of the paper are outlined as follows:
1. Improving the performance of original BES algorithm by enhancing and modifying it with Quasi opposition-based learning.
2. Proposed a hybrid model, LSTM-MBES-Adaboost to enhance the accuracy and provide more optimum solution when predicting concrete compressive strength.

3. The preprocessed concrete compressive strength dataset was evaluated in comparison to the proposed method (LSTM-MBES-Adaboost) model, LSTM model, LSTM-BES model, and LSTM-MBES model.

2. Related Works

There are some related works which related to hybrid of AdaBoost and LSTM. A study [6], it uses ensemble learning approach between AdaBoost and LSTM to predict sea surface temperature. The rolling prediction scheme was used to find patterns in sea surface temperature based on preceding values. Averaging strategy was implemented to combine and average both predictions result from LSTM and AdaBoost in order to obtain final prediction result. Overall, it was able to improve the RMSE values of the predictions in most of the prediction horizons and locations. However, in certain locations the RMSE starts to increase due to variations and seasonality happens to the sea surface temperature. Another study [7], proposed AdaBoost-LSTM ensemble learning approach by combining both AdaBoost and LSTM to predict financial time series. It produces 19.44% - 22.33% better directional forecasts than single models. It also significantly improves the performance and reaches the accuracy 76.88% for exchange rates forecasting.

Several researches have researched some existed works related with Bald Eagle Search and LSTM. A study [9], proposed to optimize the LSTM through modified Bald Eagle Search algorithm to predict wind power. It modifies the selection phase of the Bald Eagle Search algorithm; where the parameter is no longer fixed value, instead it is being optimized. Empirical model decomposition (EMD) method is used to decompose the SCADA wind power datasets into several parts. LSTM was trained using the hyperparameters obtained from the MBES. Results have shown that the proposed method has positive impacts in optimizing the LSTM parameters and EMD has significant effect in the data preprocessing. However, it is using the historical datasets in the model; while wind power is constantly changing depending on environment factors. Another study [10], proposed VMD-IBES-LSTM model to forecast aquatic product price in China. Variational Model Decomposition (VMD) is used to decompose the non-stationary time series data. Optimization in Bald Eagle Search is performed through adding tent mapping strategy in the population initialization stage of Bald Eagle Search and levy flight is added to perform local search strategy. The parameters obtained from IBES is used to optimize the LSTM. This hybrid model managed to outperform other models in terms of MSE, MAE, MAE, and MAPE indicators. However, some errors in accuracy happen due to fluctuations in the price and the standards in data collection should be improved when building the models.

There are some related works which utilizes the Quasi opposition-based learning in different fields. In the study [15], proposed to incorporate Quasi opposition-based learning and dimensional search strategy to enhance Lightning Attachment Procedure Optimization (LAPO). To prevent falling into local optima conditions and to improve the convergence rate, Quasi opposition-based learning is proposed. In addition, dimensional search strategy is proposed because the changes of variable in one dimension can affect another dimension which results in poor convergence rate. Results show that the proposed model outperforms the performance in terms of accuracy and convergence rate of the other state of arts algorithms. However, it is quite sensitive to number of dimensions in the dimension search strategy. The higher the number of dimensions, will affect the global search results. Another study [16], optimized the whale algorithm through chaos mapping and Quasi opposition-based learning. Chaos mapping is used to generate initial population and improve the convergence rate. Quasi opposition-based learning is used to ensure it doesn’t trap in local optima solutions. It results in a better performance in terms of accuracy and convergence speed. However, the running time is longer compared to other meta-heuristics algorithms.

3. Proposed Methodology

Hybrid LSTM-MBES-AdaBoost

The method proposed in this study is a hybrid model of LSTM-MBES-AdaBoost to predict the concrete compressive strength. The modified Bald Eagle Search algorithm is used to optimize the LSTM parameters. Quasi opposition-based learning is incorporated to modify and improve the performance of Bald Eagle Search algorithm. In addition, hybridization of AdaBoost with optimized LSTM model is performed as the final step to obtain final best solution.

Figure 1 shows the illustration of the overview of proposed algorithm. In the Figure 1, initially the pre-processing steps are applied to the concrete compressive strength dataset. The modification performed to Bald Eagle Search algorithm is done by incorporated Quasi opposition-based learning into the Bald Eagle Search algorithm during the population initialization stage when determine the best solution. Results obtained from the Bald Eagle Search algorithm will be compared with the results obtained after using Quasi opposition-based learning in the population initialization stage to determine the initial best solution.

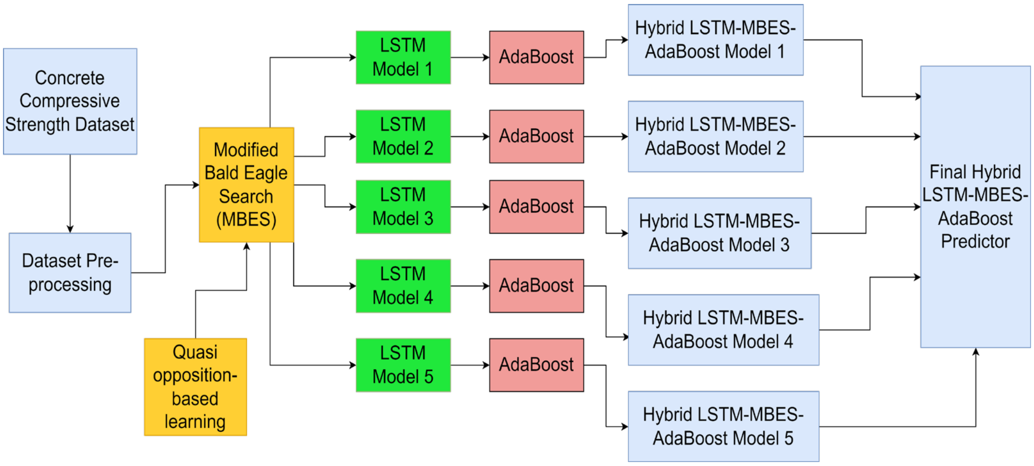

Figure 1. Overview of Proposed Algorithm

Following are the components involved in the proposed algorithm:

1. AdaBoost

AdaBoost or usually known as adaptive boosting algorithm, is mainly used to enhance the performance of previous boosting algorithm [17]. The main idea of AdaBoost is to maintain a set of weight distribution in the training samples. In each cycle, weight of training samples which are misclassified by the weak learners will be increased, so that they can learn from the hard samples. While the training samples which are correctly classified, the weight will be decreased [18]. Following figure shows the architecture diagram of AdaBoost:

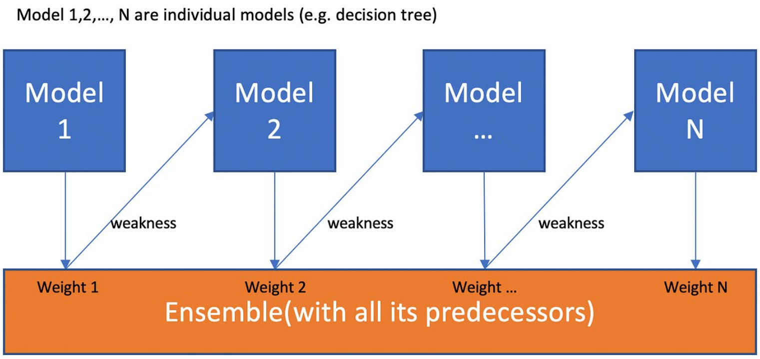

Figure 2. AdaBoost Architecture Diagram [19]

In the Figure 2, the weakness and error from the first model are taken as the input to the next model. The next consecutive N models are taking errors from previous models as an input. These models can be regarded as decision tree. This process keeps on repeating until the errors are minimized.

2. LSTM

Long short-term memory (LSTM) is one of deep learning models in the form of recurrent neural network which has the capability to learn long time dependencies, proposed by Hochreiter, S., & Schmidhuber, J. [20]. It consists of four main components which are input gate, forget gate, output gate, and memory cell. The memory cell contains self-recurrent connection, while forget allows memory cell to forget previous information, and output gate is to obtain the final result [21].

Figure 3 shows the architecture diagram of LSTM. In the Figure 3, several chunks of neural network are labelled as A. The input states into the network are labelled as X. Label h indicates the output value of the network. Inside the network, it consists three gates of LSTM which are input gate, forget gate, and output gate.

Input gate is used to determine which information will be passed into the memory cell based from previous output and the importance. Following shows the equation involved in input gate:

$$\text{Input gate: } i_t = \sigma\,(X_t * U_i + H_{t-1} + W_i) \tag{1}$$

Where i_t is the input at current timestamp t, U_i is the weight of matrix input, H_{t-1} is the hidden state at the previous timestamp, W_i is the weight matrix of input associated with hidden state. Sigmoid function denoted by σ is also used to determine allowing 0 or 1 values to pass through the network.

Forget gate is used to determine which information will be kept from the previous timestamps. Following shows the equation involved in forget gate:

$$\text{Forget gate: } f_t = \sigma\,(X_t * U_f + H_{t-1} * W_f) \tag{2}$$

Where X_t is the input to the current timestamp, U_f is the weight associated with the input, H_{t-1} is the hidden state from previous timestamp, W_f is the weight matrix associated with hidden state. This is also decided by sigmoid function which is denoted by σ; it will omit the states if input X_t produces 0 value and will keep the states the value of input X_t is 1.

Output gate is used to determine which value will be returned as the hidden state in the next time steps. Following shows the equation involved in output gate:

Output gate: $o_t = \sigma(X_t * U_o + H_{t-1} + W_0)$ (3)

Where o_t is the output at the current timestamp t, U_o is the weight of matrix, H_{t-1} is the hidden state at the previous timestamp, W_o is the weight matrix of input associated with output. Sigmoid function denoted by σ is also used to determine allowing 0 or 1 values to pass through the network.

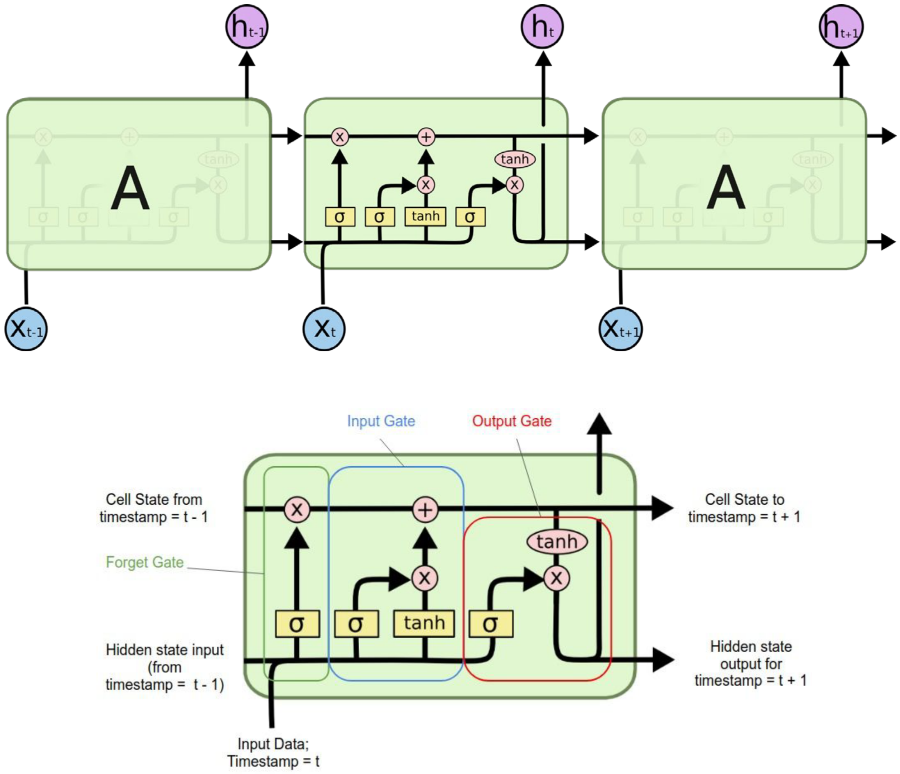

Figure 3. LSTM Architecture Diagram [22][23]

3. Bald Eagle Search

Proposed by Alsattar H.A., Zaidan A.A, and Zaidan B.B [24], bald eagle search is one of the metaheuristics algorithms which aims to mimic the behaviour of an eagle when targeting a prey. Figure 4 illustrates the stages involved in the Bald Eagle Search algorithm.

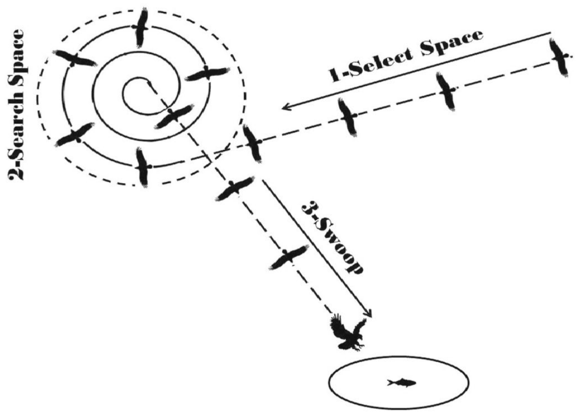

Figure 4. Stages in Bald Eagle Search Algorithm [24]

In the Figure 4, it shows the Bald Eagle Search algorithm consists of three main stages which are select stage, search stage, and swoop stage.

a. Select stage

Bald eagle will randomly select space to find the area that has the most amount of preys. Following shows the equation in select stage:

$$Pnew, i = Pbest + \propto * r\ (Pmean - Pi) \tag{4}$$

Where $Pnew$ is the current position of the bald eagle during i^{th} iteration and i is the current number of iteration. $Pbest$ is the current most optimal position obtained by the bald eagle through previous searches. $\propto$ is a parameter to control the changes in position with having the range from 1.5 to 2. While r is a random number within the range of 0 to 1. $Pmean$ is obtained from the average distribution of bald eagle's position after the previous searches.

b. Search stage

Bald eagle starts to search within the space through flying in conical spirals and accelerate the speed. Polar coordinates is used to plot the movements of bald eagle. Following shows the equation involved in search stage:

$$\theta(i) = \alpha * \pi * random(0,1) \tag{5}$$
$$r(i) = \theta(i) + R * random(0,1) \tag{6}$$

In equation 5, the polar angle of conical spiral is defined as $\theta(i)$. α value spans from 0 to 5. While in equation 6, $r(i)$ is the polar diameter of the conical spiral. The R value range from 0.5 to 2 which determines the number of search cycle.

$xr(i) = r(i) * \sin(\theta(i)),$

$$yr(i) = r(i) * \cos\ (\theta(i)) \tag{7}$$
$$x(i) = \frac{xr(i)}{\max(|xr|)},\quad y(i) = \frac{yr(i)}{\max(|yr|)} \tag{8}$$

Equation 7 defines the calculation involved to calculate the polar angle between both axes, x and y. The polar coordinates are represented in the form of $xr(i)$ and $yr(i)$. While equation 8 shows the derivation formula to obtain $x(i)$ and $y(i)$ values.

$$Pi, new = Pi + y(i) * (Pi - P_{i+1}) + x(i) * (Pi - Pmean) \tag{9}$$

The final equation of search stage is shown in equation 9; where it takes the difference between current and center of polar.

c. Swoop stage

After bald eagle has searched within the space, it will swing from the best position and fly towards the prey to capture it. Following shows the equation in the swoop stage:

$Pi, new = random\ (0,1) * Pbest +$

$$xl(i) * (Pi - c1 * Pmean) + yl(i) * (Pi - c2 * Pbest) \tag{10}$$
$$xl(i) = \frac{xr(i)}{\max(|xr|)},\ yl(i) = \frac{yr(i)}{\max(|yr|)} \tag{11}$$

In the equation 10, $c1$ and $c2$ are in the range of 1 to 2; which denotes the best position and center position. Equation 11 shows the derivation formula to obtain $x(i)$ and $y(i)$ values.

4. Quasi Opposition-Based Learning

To enhance the performance of opposition-based learning, a new variation named Quasi opposition-based learning is proposed by Tizhoosh, H. R. [25]. This method uses quasi opposite points and has the same steps as opposition-based learning; which are initializing the population and generation jumping. It is proved that using quasi opposite points results in a closer solution compared to opposite points. Following shows the equation of Quasi opposition-based learning:

$$x^{qo} = rand(c, x^0) \quad , c = \frac{a+b}{2} \tag{12}$$

In the equation 12, it shows the equation of Quasi opposition-based learning; specifically, for the Quasi opposition number. Where x is a real number within the interval of $x \in [\mathrm{a}, \mathrm{b}]$. c is the center interval value between a and b. $rand(c, x^0)$ is the random number distributed between center point and search space.

$$x_i^{qo} = rand(c_i, x_i^0) \quad , c_i = \frac{a_i+b_i}{2} \quad , \; i = 1, 2, \ldots, \mathrm{n} \tag{13}$$

In the equation 13, it shows the equation of Quasi opposition-based learning; specifically, for the Quasi opposition point. Where $P(x_1, x_2, \ldots, x_n)$ defined as the point in n dimensional space. x_i is a real number within the interval of $x_i \in [\mathrm{a_i}, \mathrm{b_i}]$. *The Quasi opposite point* $\mathrm{QP^0}$ *is represented by its coordinates* $(\mathrm{x_1^{q0}}, \mathrm{x_2^{q0}}, \ldots, \mathrm{x_n^{q0}})$.

A. Proposed Method

Following shows the equation of Bald Eagle Search population initialization method:

$$x_i = lowerbound + (upperbound - lowerbound) * rand \tag{14}$$

Where $lowerbound$ is the lower limit value determined in the problem space and $upperbound$ is the upper limit value determined in the problem space. $rand$ is a random number generated in the given space. x_i is the solution at the current i-th iteration.

Following shows the equation of Quasi opposition-based learning population initialization method:

$$curValue = lowerbound + upperbound - x_i \tag{15}$$

$$curMean = \frac{lowerbound+upperbound}{2} \tag{16}$$

Where $lowerbound$ is the lower limit value determined in the problem space and $upperbound$ is the upper limit value determined in the problem space. $rand$ is a random number generated in the given space. x_i is the solution at the current i-th iteration which is obtained from Bald Eagle Search population initialization method. $curValue$ is the current value and the calculation is shown in equation 15. While $curMean$ refers to the average between $lowerbound$ and $upperbound$.

$$quasiSolution = curMean + (curValue - curMean) * rand(0,1) \; ; \quad if \; curValue < curMean \tag{17}$$

$$quasiSolution = curValue + (curMean - curValue) * rand(0,1); \quad if \; curValue \geq curMean \tag{18}$$

In equation 17 and equation 18, it shows that there are two conditions that need to be prechecked to determine and update final solution of Quasi opposition-based learning. Specifically, equation 17 is used to obtain the $quasiSolution$ if the condition

$curValue < curMean$ is fulfilled. While equation 18 is used to obtain the $quasiSolution$ if the condition $curValue \geq curMean$ is fulfilled.

Quasi opposition-based learning is chosen to modify the Bald Eagle Search algorithm as it is able to improve the convergence rate and exploration ability of Bald Eagle. Quasi opposition-based learning generates the solution by using origin population solution obtained from Bald Eagle Search. Thus, it will enhance the diversity of the population and improve convergence rate. In addition, the population generated by both Bald Eagle Search population initialization method and quasi opposition-based learning method will be compared through training in LSTM model. The solution that has the least fitness value will be chosen and updated as the initial best solution.

In addition, all of the candidates of best solution obtained in the end of each iteration will be fed into the LSTM base model. In total, there are 5 LSTM models generated with the candidates of best solution obtained from MBES. Hybridization will be performed between the LSTM models and AdaBoost model to obtain the final best solution and best fitness value. All of the LSTM models which obtained the optimized parameters through MBES will be incorporated with AdaBoost to further predict. The fitness cost value will be evaluated and compared between each of the LSTM models which have been performed hybridization with AdaBoost. Thus, the combination of LSTM and AdaBoost algorithm will result in a hybrid model which serves as the final predictor model.

The flowchart of proposed algorithm hybrid model of LSTM-MBES-AdaBoost is shown in Figure 5 along with the explanations steps.

1. Set the parameters of Bald Eagle Search algorithm which consist of dimension number, population size, max iteration, upper bound, lower bound, alpha, A_factor, R_factor, c1, and c2.
2. Preprocess the dataset and split to ratio of 80% for training and 20% for testing. Construct LSTM model and obtained LSTM parameters from BES model will be fed into the LSTM model to calculate fitness function.
3. Initialize best solution using population initialization method from BES and compared with Quasi opposition-based learning. Calculate fitness function in each of the iteration during this stage. The solution with least fitness value, will be updated as the best solution.
4. Based from the number of max iteration given, the stages of Bald Eagle Search algorithm are performed. In the select stage, each position of the solution is updated according to Equation 4. Fitness value is calculated by feeding the obtained solution into the LSTM model to train and update the best solution.
5. In search stage, each position of the solution is updated according to Equation 9. Fitness value is calculated by feeding the obtained solution into the LSTM model to train and update the best solution.
6. In swoop stage, each position of the solution is updated according to Equation 11. Fitness value is calculated by feeding the obtained solution into the LSTM model to train and update the best solution.
7. Store the candidates of best solution obtained in each iteration into an array.
8. Perform hybridization with AdaBoost and fed all of the best solution candidates into the LSTM model. Compare the result between each candidate and calculate the fitness value.
9. Output the final best LSTM parameters with the best fitness value.

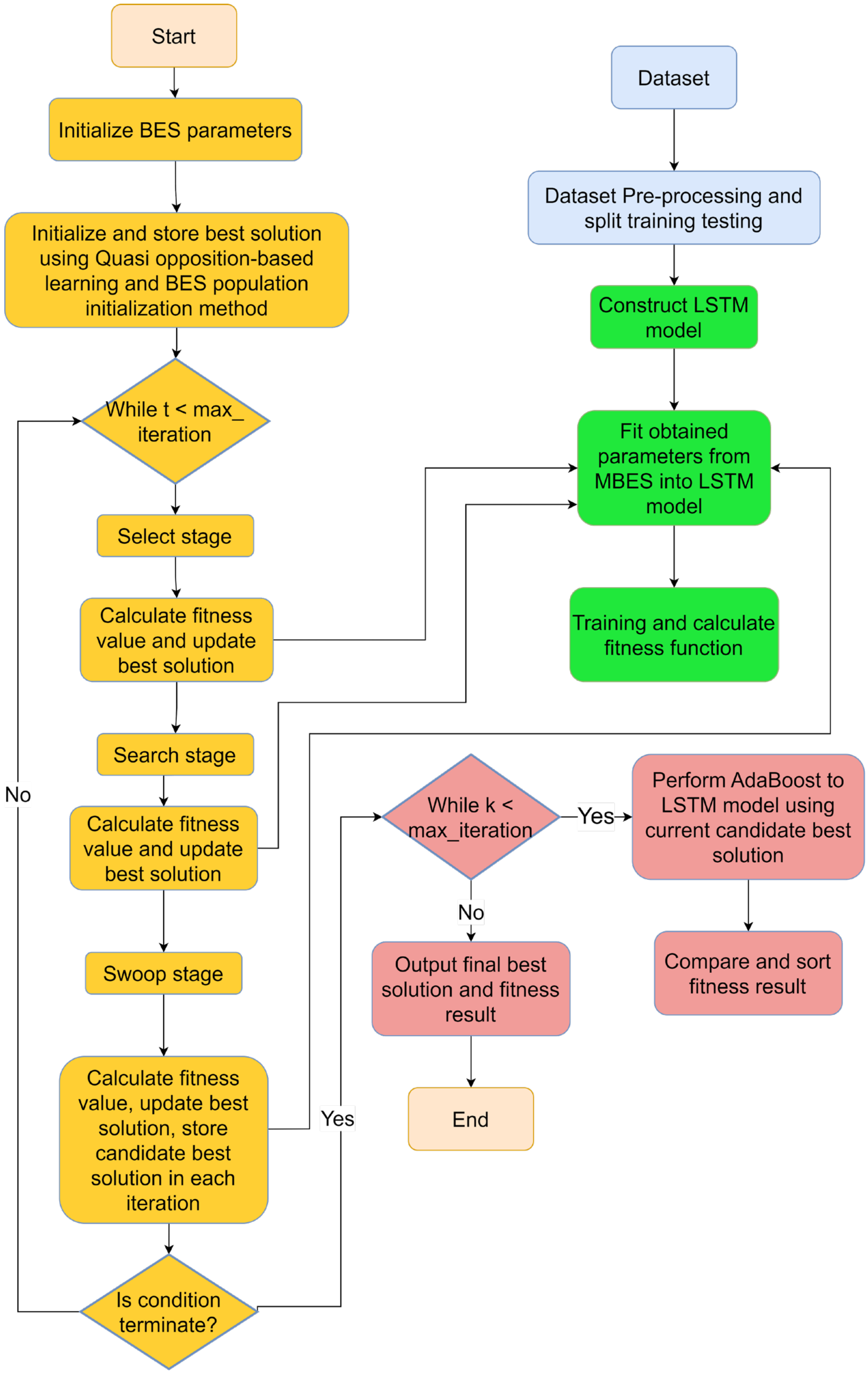

Figure 5. Flowchart of LSTM-MBES-AdaBoost model

Following shows the pseudocode of the proposed algorithm Hybrid of LSTM-MBES-AdaBoost.

Algorithm 1 Pseudocode of Hybrid LSTM-MBES-AdaBoost

```
Initialize parameters of BES
while i < population size do
// Generate best solution using BES population initialization method
  x[i] = lowerbound + (upperbound – lowerbound) * random
// Generate best solution using Quasi opposition-based learning using Equation 12 – 13
for k = 0, 1, …. dimensionNumber do
curValue = lowerbound + upperbound – x[k]
curMean = (lowerbound + upperbound) / 2
if curValue < curMean then
quasiValue = curMean + (curValue – curMean) * rand(0,1)
else
quasiValue = curValue + (curMean – curValue) * rand(0,1)
end if
end for
if fitnessFunction(Quasi solution) < fitnessFunction(BES solution) then
Update best solution is Quasi solution
else
Update best solution is BES solution
end if
end while
while t < max iteration do
Calculate fitness function and update solution for select stage using Equation 4
Calculate fitness function and update solution for search stage using Equation 5 – 9
Calculate fitness function and update solution for swoop stage using Equation 10 – 11
Store best solution candidate into an array
end while
while k < max iteration do
Perform AdaBoost and fit the current candidate solution x[k] into LSTM model to train
Compare and sort the result
end while
Output final best solution LSTM parameter and best fitness cost
```

5. Experiment and Results

A. Dataset and Preprocessing

The concrete compressive strength dataset used in this study is obtained from UCI Machine Learning Repository [11]. It has 1030 samples of data and 9 attributes. Following is the list of attributes in the dataset:

a. Cement: quantity of cement in kg/m^3 used in the mixture
b. Blast furnace slag: quantity of blast furnace slag admixtures in kg/m^3 used in the mixture
c. Fly ash: quantity of fly ash admixtures in kg/m^3 used in the mixture
d. Water: quantity of water in kg/m^3 used in the mixture
e. Superplasticizer: quantity of superplasticizer admixtures in kg/m^3 used in the mixture
f. Coarse aggregate: quantity of coarse aggregate in kg/m^3 used in the mixture
g. Fine aggregate: quantity of fine aggregate in kg/m^3 used in the mixture
h. Age: the age of concrete since casting date
i. Concrete compressive strength: the maximum compressive strength that can be achieved by the concrete after casting period.

Following Table 1 shows the five samples data of the dataset:

Table 1. Sample Data of Concrete Compressive Dataset

Cement (kg in a m^3 mixture)	Blast Furnace Slag (kg in a m^3 mixture)	Fly Ash (kg in a m^3 mixture)	Water (kg in a m^3 mixture)	Superpla sti-cizer (kg in a m^3 mixture)	Coarse Aggreg-ate (kg in a m^3 mixture)	Fine Aggreg-ate (kg in a m^3 mixture)	Age (day)	Concrete compress-ive strength (MPa, megapascals)
540,0	0,0	0,0	162,0	2,5	1040,0	676,0	28	79,99
540,0	0,0	0,0	162,0	2,5	1055,0	676,0	28	61,89
332,5	142,5	0,0	228,0	0,0	932,0	594,0	270	40,27
332,5	142,5	0,0	228,0	0,0	932,0	594,0	365	41,05
198,6	132,4	0,0	192,0	0,0	978,4	825,5	360	44,30

There are no missing values in the dataset and all of the values are in the correct form. Normalization is performed to scale all of the data into range of 0 to 1. The dataset is split into 80% for training and 20% for testing.

B. Evaluation

To evaluate the performance of the proposed algorithm, experiments are being performed to compare between LSTM, LSTM-BES, LSTM-MBES, and LSTM-MBES-AdaBoost models. Metrics used to evaluate consist of Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). Following shows the equation for each metric:
Mean Squared Error (MSE); defines the squared errors obtained, the larger the value indicates the larger the error.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (19)$$

Where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of data points.
Mean Absolute Error (MAE); defines the actual errors obtained by the model; the larger the value indicates the larger the error.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|^2 \quad (20)$$

Where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of data points.
Mean Absolute Percentage Error (MAPE); defines the prediction accuracy; the smaller the value indicates the model is more accurate.

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{y_i} \quad (21)$$

Where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of data points.

C. Experiment Results

The concrete compressive strength dataset obtained from UCI Machine Learning Repository [11] is used as the testing dataset. Experiments are being performed by comparing the performance between LSTM, LSTM-BES, LSTM-MBES, and LSTM-MBES-AdaBoost model. Metrics to evaluate the testing results are based on Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).
To make it uniform during the testing, all of the testing in each model is executed in 5 trials and all of the models are using similar testing parameters configuration as shown in Table 2. Table 2 shows the parameters used during testing.

Table 2. Testing Parameters Configuration of All Models

Method	Parameter	Value
LSTM	Number of execution or repetitions	5
	Number of epochs	50
	Number of layers	3
	Optimizer	Adam
LSTM-BES, LSTM-MBES, LSTM-MBES-AdaBoost	Number of execution or repetitions	5
	Number of epochs	50
	Number of layers	3
	Optimizer	Adam
	BES A_factor	8
	BES R_factor	1.5
	BES number of dimensions	3
	BES population size	5
	BES max iteration	5
	BES c1, c2	2
	BES α factor	2
	BES lower bound	0.01
	BES upper bound	300

- *LSTM Model*

To perform testing on LSTM model which serves as the baseline, trial and error are being performed to determine the parameters configuration of the LSTM model. Following Table 3 shows the configuration used in each trial and the corresponding evaluations based on the predictions. The values in "Configuration" column, represents the number of units in first layer, number of units in dropout layer, and number of units in second layer of LSTM model respectively. The values in "Configuration" column represents the LSTM parameters used to train the LSTM model in each of the trial respectively. From Table 3, it can be observed that; trial 2 and trial 5 exhibits the least MSE, MAE, and MAPE compared to the others.

Table 3. Configuration and Error Analysis of LSTM Model

Trial	Configuration	MSE	MAE	MAPE
1	(150, 0.2, 150)	0.01379	0.09416	0.29377
2	(300, 0.2, 100)	0.01024	0.08131	0.25930
3	(180, 0.2, 100)	0.01198	0.08788	0.31779
4	(150, 0.1, 150)	0.01222	0.08921	0.30949
5	(300, 0.1, 100)	0.01052	0.08139	0.25738

- *LSTM-BES Model*

LSTM-BES model is constructed using the original Bald Eagle Search algorithm [24] to obtain the optimum parameters as solution to be fed into the LSTM model. Following Table 4 shows the optimization results obtained by LSTM-BES model. "Initial Best Solution and Cost" column refers to the solution and costs that are obtained during population initialization stage when determine best solution in each iteration. "Best Solution Obtained and Cost" column refers to the final best solution and cost which is obtained at the end of each iteration after perform all of the BES stage (select stage, search stage, and swoop stage). The solutions refer to the number of units in first layer, dropout layer, and second layer of LSTM respectively.

Costs refer to the fitness function cost which are obtained during evaluating the MSE in the iteration process of LSTM-BES model.

Table 4. Optimization Results – LSTM-BES Model

Trial	Iteration	Initial Best Solution and Cost	Best Solution Obtained and Cost
1	1	Solution: (201, 0.7309374369863978, 49) Cost: 0.014727532619027123	Solution: (164, 0.10929401643078726, 229) Cost: 0.01240090596275544
	2	Solution: (201, 0.7309374369863978, 49) Cost: 0.014727532619027123	Solution: (126, 0.23791079821850078, 300) Cost: 0.01215461519010562
	3	Solution: (201, 0.7309374369863978, 49) Cost: 0.014727532619027123	Solution: (122, 0.036703894377200494, 186) Cost: 0.01145821906447492
	4	Solution: (128, 0.2681132036186432, 83) Cost: 0.012895263873886995	Solution: (122, 0.036703894377200494, 186) Cost: 0.01145821906447492
	5	Solution: (128, 0.2681132036186432, 83) Cost: 0.012895263873886995	Solution: (122, 0.036703894377200494, 186) Cost: 0.01145821906447492
2	1	Solution: (193, 0.035432734431842416, 24) Cost: 0.012078467362748835	Solution: (283, 0.01, 30) Cost: 0.00957533687078179
	2	Solution: (193, 0.035432734431842416, 24) Cost: 0.012078467362748835	Solution: (283, 0.01, 30) Cost: 0.00957533687078179
	3	Solution: (193, 0.035432734431842416, 24) Cost: 0.012078467362748835	Solution: (283, 0.01, 30) Cost: 0.00957533687078179
	4	Solution: (193, 0.035432734431842416, 24) Cost: 0.012078467362748835	Solution: (283, 0.01, 30) Cost: 0.00957533687078179
	5	Solution: (193, 0.035432734431842416, 24) Cost: 0.012078467362748835	Solution: (283, 0.01, 30) Cost: 0.00957533687078179
3	1	Solution: (217, 0.5836505090137305, 256) Cost: 0.014823510846771814	Solution: (252, 0.01, 15) Cost: 0.010839684747670783
	2	Solution: (232, 0.39861648444673636, 60) Cost: 0.013026405852776389	Solution: (270, 0.15659195759178, 54) Cost: 0.010689436600127401
	3	Solution: (232, 0.39861648444673636, 60) Cost: 0.013026405852776389	Solution: (270, 0.15659195759178, 54) Cost: 0.010689436600127401
	4	Solution: (232, 0.39861648444673636, 60) Cost: 0.013026405852776389	Solution: (270, 0.15659195759178, 54) Cost: 0.010689436600127401

Trial	Iteration	Initial Best Solution and Cost	Best Solution Obtained and Cost
	5	Solution: (232, 0.39861648444673636, 60) Cost: 0.013026405852776389	Solution: (264, 0.11141260306885199, 33) Cost: 0.010161756409785748
4	1	Solution: (15, 0.20990330865524306, 219) Cost: 0.014558387532439595	Solution: (228, 0.05951216393951437, 235) Cost: 0.011309303393676542
	2	Solution: (15, 0.20990330865524306, 219) Cost: 0.014558387532439595	Solution: (300, 0.1566789030464 2518, 184) Cost: 0.010367076812011355
	3	Solution: (15, 0.20990330865524306, 219) Cost: 0.014558387532439595	Solution: (300, 0.17996808182602642, 176) Cost: 0.009610806740411468
	4	Solution: (15, 0.20990330865524306, 219) Cost: 0.014558387532439595	Solution: (300, 0.17996808182602642, 176) Cost: 0.009610806740411468
	5	Solution: (240, 0.32386670360175457, 132) Cost: 0.012245201418585216	Solution: (300, 0.17996808182602642, 176) Cost: 0.009610806740411468
5	1	Solution: (40, 0.4302009965592483, 200) Cost: 0.013812601157167276	Solution: (278, 0.09004425762971778, 60) Cost: 0.010680153037626426
	2	Solution: (40, 0.4302009965592483, 200) Cost: 0.013812601157167276	Solution: (258, 0.01, 50) Cost: 0.009684773833811312
	3	Solution: (155, 0.12500698841857122, 58) Cost: 0.012496455702195144	Solution: (258, 0.01, 50) Cost: 0.009684773833811312
	4	Solution: (207, 0.4215579916673586, 185) Cost: 0.012404453589455492	Solution: (258, 0.01, 50) Cost: 0.009684773833811312
	5	Solution: (207, 0.4215579916673586, 185) Cost: 0.012404453589455492	Solution: (258, 0.01, 50) Cost: 0.009684773833811312

From the above Table 4, it can be observed that the best optimum result using LSTM-BES model can be found in trial 2 having cost 0.012078467362748835; which is the lowest among the others. The lower the amount of cost, indicates the better performance of that trial. Highest cost can be found in trial 3 with cost 0.013026405852776389 and best solution cost 0.010161756409785748.

Following Table 5 shows the parameters configuration which are used during testing and analysis. Testing are being performed on the concrete compressive strength dataset [11] through LSTM model which uses the parameters configuration in each of the trial as stated in Table 4. The parameters configuration used in Table 5 is obtained from the best solution obtained in each iteration which located at the column "Best Solution Obtained and Cost" as described in Table 4.

Table 5. Configurations and Error Analysis – LSTM BES Model

Trial	Iteration	Configuration	MSE	MAE	MAPE
1	1	(164, 0.10929401643078726, 229)	0.01244	0.09042	0.32855
	2	(126, 0.23791079821850078, 300)	0.01169	0.08766	0.29988
	3	(122, 0.036703894377200494, 186)	0.01224	0.08807	0.30712
	4	(122, 0.036703894377200494, 186)	0.01224	0.08807	0.30712
	5	(122, 0.036703894377200494, 186)	0.01224	0.08807	0.30712
2	1	(283, 0.01, 30)	0.01116	0.08416	0.28185
	2	(283, 0.01, 30)	0.01116	0.08416	0.28185
	3	(283, 0.01, 30)	0.01116	0.08416	0.28185
	4	(283, 0.01, 30)	0.01116	0.08416	0.28185
	5	(283, 0.01, 30)	0.01116	0.08416	0.28185
3	1	(252, 0.01, 15)	0.01167	0.08720	0.34898
	2	(270, 0.15659195759178, 54)	0.01119	0.08738	0.33416
	3	(270, 0.15659195759178, 54)	0.01119	0.08738	0.33416
	4	(270, 0.15659195759178, 54)	0.01119	0.08738	0.33416
	5	(264, 0.11141260306885199, 33)	0.01308	0.09040	0.28038
4	1	(228, 0.05951216393951437, 235)	0.01131	0.08431	0.278014
	2	(300, 0.15667890304642518, 184)	0.01120	0.08459	0.31663
	3	(300, 0.17996808182602642, 176)	0.01126	0.08343	0.25294
	4	(300, 0.17996808182602642, 176)	0.01126	0.08343	0.25294
	5	(300, 0.17996808182602642, 176)	0.01126	0.08343	0.25294
5	1	(278, 0.09004425762971778, 60)	0.01153	0.08665	0.29665
	2	(258, 0.01, 50)	0.01121	0.08299	0.26692
	3	(258, 0.01, 50)	0.01121	0.08299	0.26692
	4	(258, 0.01, 50)	0.01121	0.08299	0.26692
	5	(258, 0.01, 50)	0.01121	0.08299	0.26692

From the results shown in Table 5, it is evident that best performance of LSTM-BES model can be found in trial 2, demonstrating lowest values for MSE, MAE, and MAPE when compared to other trials. Specifically, in trial 2, the model achieved MSE of 0.01116, MAE of 0.08416, and MAPE of 0.28185, indicating better prediction result compared to other trials.

- *LSTM-MBES Model*

LSTM-MBES model incorporates the Modified Bald Eagle Search (MBES) algorithm which is being modified by Quasi Opposition Based Learning. This combination is utilized to find optimum solution as the parameters in LSTM model and to compare with LSTM-BES model for its performance.

Following Table 6 shows the optimization results obtained by LSTM-MBES model. “Initial Best Solution and Cost” column refers to the solution and costs that are obtained during population initialization stage when determine best solution in each iteration. “Best Solution Obtained and Cost” column refers to the final best solution and cost which is obtained at the end of each iteration after perform all of the MBES stage (select stage, search stage, and swoop stage). The solutions refer to the number of units in first layer, dropout layer, and second layer of LSTM respectively. Costs refer to the fitness function cost which are obtained during evaluating the MSE in the iteration process of LSTM-MBES model.

Table 6. Optimization Results – LSTM-MBES Model

Trial	Iteration	Initial Best Solution and Cost	Best Solution Obtained and Cost
1	1	Solution: (200, 0.26316710184489045, 189) Cost: 0.012796518368216854	Solution: (258, 0.13406640751780996, 116) Cost: 0.009875870977048909
	2	Solution: (226, 0.1000891633288743, 112) Cost: 0.01028562583791258	Solution: (258, 0.13406640751780996, 116) Cost: 0.009875870977048909
	3	Solution: (226, 0.1000891633288743, 112) Cost: 0.01028562583791258	Solution: (258, 0.13406640751780996, 116) Cost: 0.009875870977048909
	4	Solution: (226, 0.1000891633288743, 112) Cost: 0.01028562583791258	Solution: (290, 0.15506803202800842, 132) Cost: 0.009874107391612232
	5	Solution: (226, 0.1000891633288743, 112) Cost: 0.01028562583791258	Solution: (264, 0.01, 39) Cost: 0.00974523229521914
2	1	Solution: (195, 0.33377698350402474, 190) Cost: 0.012564597848626652	Solution: (192, 0.2186141982031269, 160) Cost: 0.011617445063851317
	2	Solution: (195, 0.33377698350402474, 190) Cost: 0.012564597848626652	Solution: (200, 0.01, 164) Cost: 0.011483010704792174
	3	Solution: (195, 0.33377698350402474, 190) Cost: 0.012564597848626652	Solution: (181, 0.01, 93) Cost: 0.01042854869514286
	4	Solution: (195, 0.33377698350402474, 190) Cost: 0.012564597848626652	Solution: (181, 0.01, 93) Cost: 0.01042854869514286
	5	Solution: (195, 0.33377698350402474, 190) Cost: 0.012564597848626652	Solution: (181, 0.01, 93) Cost: 0.01042854869514286

Trial	Iteration	Initial Best Solution and Cost	Best Solution Obtained and Cost
3	1	Solution: (166, 0.8185149902075414, 279) Cost: 0.013732854450158701	Solution: (127, 0.01, 207) Cost: 0.011735211655477676
	2	Solution: (217, 0.4270953107915287, 151) Cost: 0.012821089198345303	Solution: (127, 0.01, 207) Cost: 0.011735211655477676
	3	Solution: (149, 0.09916541444704174, 227) Cost: 0.012466839462988447	Solution: (141, 0.01, 109) Cost: 0.011188227321862712
	4	Solution: (149, 0.09916541444704174, 227) Cost: 0.012466839462988447	Solution: (157, 0.01, 58) Cost: 0.010763918051138565
	5	Solution: (149, 0.09916541444704174, 227) Cost: 0.012466839462988447	Solution: (158, 0.015206723419620342, 158) Cost: 0.010646464041867925
4	1	Solution: (226, 0.37181263831468825, 154) Cost: 0.013589569503033517	Solution: (149, 0.3790370241069617, 185) Cost: 0.012077340317700346
	2	Solution: (226, 0.37181263831468825, 154) Cost: 0.013589569503033517	Solution: (104, 0.01, 252) Cost: 0.011590346459122231
	3	Solution: (240, 0.47835291893808257, 91) Cost: 0.01328700096393999	Solution: (142, 0.01, 300) Cost: 0.011586575651898606
	4	Solution: (163, 0.07039592409096747, 258) Cost: 0.012836883141284922	Solution: (133, 0.01, 300) Cost: 0.010813835644464672
	5	Solution: (150, 0.40664571496867286, 179) Cost: 0.012440808901462382	Solution: (133, 0.01, 300) Cost: 0.010813835644464672
5	1	Solution: (180, 0.4120518119398925, 259) Cost: 0.013060780177008603	Solution: (217, 0.043700556915905936, 132) Cost: 0.0105567443061006
	2	Solution: (180, 0.4120518119398925, 259) Cost: 0.013060780177008603	Solution: (245, 0.06711297795558117, 125) Cost: 0.009336871583671604
	3	Solution: (203, 0.20778613947859773, 144) Cost: 0.012024145534009234	Solution: (245, 0.06711297795558117, 125) Cost: 0.009336871583671604
	4	Solution: (203, 0.20778613947859773, 144) Cost: 0.012024145534009234	Solution: (245, 0.06711297795558117, 125) Cost: 0.009336871583671604
	5	Solution: (203, 0.20778613947859773, 144) Cost: 0.012024145534009234	Solution: (245, 0.06711297795558117, 125) Cost: 0.009336871583671604

From the above Table 6, it can be observed that the best performance achieved by LSTM-MBES model happens in trial 1. This is indicated with initial best solution and cost with 0.01028562583791258 which is improved from 0.012796518368216854; having reduction 19.62% from the initial point value. In addition, trial 5 managed to obtain best solution and cost with 0.009336871583671604 which is the lowest among other trials. The provided results have shown that by modifying the bald eagle search algorithm using Quasi Opposition Based Learning, can slightly improve the convergence rate. Specifically, the outcomes achieved by LSTM-MBES model, able to obtain lower initial cost for the best solution and cost compared to LSTM-BES model. This implies that LSTM-MBES able to provide more optimize solution compared to LSTM-BES model.

Following Table 7 shows the parameters configuration which are used during testing and analysis. Testing are being performed on the concrete compressive strength dataset [11] through LSTM model which uses the parameters configuration in each of the trial as stated in Table 6. The parameters configuration used in Table 7 is obtained from the best solution obtained in each iteration which located at the column “Best Solution Obtained and Cost” as described in Table 6.

Table 7. Configuration and Error Analysis – LSTM-MBES Model

Trial	Iteration	Configuration	MSE	MAE	MAPE
1	1	(258, 0.13406640751780996, 116)	0.01091	0.08426	0.28779
	2	(258, 0.13406640751780996, 116)	0.01091	0.08426	0.28779
	3	(258, 0.13406640751780996, 116)	0.01091	0.08426	0.28779
	4	(290, 0.15506803202800842, 132)	0.01001	0.08008	0.27867
	5	(264, 0.01, 39)	0.01012	0.08143	0.28442
2	1	(192, 0.2186141982031269, 160)	0.01232	0.08916	0.29046
	2	(200, 0.01, 164)	0.01069	0.08332	0.27936
	3	(181, 0.01, 93)	0.01127	0.08423	0.28594
	4	(181, 0.01, 93)	0.01127	0.08423	0.28594
	5	(181, 0.01, 93)	0.01127	0.08423	0.28594
3	1	(127, 0.01, 207)	0.01244	0.08840	0.30331
	2	(127, 0.01, 207)	0.01244	0.08840	0.30331
	3	(141, 0.01, 109)	0.01274	0.08986	0.33281
	4	(157, 0.01, 58)	0.01246	0.08828	0.30167
	5	(158, 0.015206723419620342, 158)	0.01121	0.08457	0.27186
4	1	(149, 0.3790370241069617, 185)	0.01193	0.08861	0.31144
	2	(104, 0.01, 252)	0.01316	0.09079	0.30864
	3	(142, 0.01, 300)	0.01283	0.08994	0.34731
	4	(133, 0.01, 300)	0.01153	0.08569	0.28697
	5	(133, 0.01, 300)	0.01153	0.08569	0.28697
5	1	(217, 0.043700556915905936, 132)	0.01163	0.08728	0.32186
	2	(245, 0.06711297795558117, 125)	0.01162	0.08476	0.26253
	3	(245, 0.06711297795558117, 125)	0.01162	0.08476	0.26253
	4	(245, 0.06711297795558117, 125)	0.01162	0.08476	0.26253
	5	(245, 0.06711297795558117, 125)	0.01162	0.08476	0.26253

From the results shown in Table 7, it is evident that the best performance of LSTM-MBES model can be found in trial 1, particularly in iteration 4. In this configuration, the model achieved lowest MSE with value 0.01001, MAE of 0.08008, and MAPE of 0.27867, indicating better result compared to other trial. In addition, this configuration has produced most favorable results when minimizing error values, compared to results obtained using LSTM-BES model. It signifies that LSTM-MBES model has demonstrated better performance in obtaining optimized solution and reducing errors compared to LSTM-BES model.

- *LSTM-MBES-AdaBoost Model*

The LSTM-MBES-AdaBoost model incorporates the optimal solution obtained from MBES stage as parameters to feed into the LSTM Model and then, hybridizes with AdaBoost. The testing of the LSTM-MBES-AdaBoost model was performed by evaluating the MSE, MAE, and MAPE using the optimal solution parameters obtained from MBES stage (as presented in Table 6). Following Table 8 shows the Configuration and Error Analysis testing on LSTM-MBES-AdaBoost model.

Table 8. Configuration and Error Analysis – LSTM-MBES-AdaBoost Model

Trial	Iteration	Configuration	MSE	MAE	MAPE
1	1	(258, 0.13406640751780996, 116)	0.00964	0.08047	0.28622
	2	(258, 0.13406640751780996, 116)	0.00964	0.08047	0.28622
	3	(258, 0.13406640751780996, 116)	0.00964	0.08047	0.28622
	4	(290, 0.15506803202800842, 132)	0.00980	0.08115	0.27518
	5	(264, 0.01, 39)	0.00963	0.08042	0.29122
2	1	(192, 0.2186141982031269, 160)	0.01129	0.08671	0.31685
	2	(200, 0.01, 164)	0.00984	0.08201	0.28924
	3	(181, 0.01, 93)	0.00895	0.07749	0.26698
	4	(181, 0.01, 93)	0.00895	0.07749	0.26698
	5	(181, 0.01, 93)	0.00895	0.07749	0.26698
3	1	(127, 0.01, 207)	0.00912	0.0795	0.28318
	2	(127, 0.01, 207)	0.00912	0.0795	0.28318
	3	(141, 0.01, 109)	0.01138	0.08692	0.30610
	4	(157, 0.01, 58)	0.01067	0.08412	0.29709
	5	(158, 0.015206723419620342, 158)	0.00927	0.08012	0.28091
4	1	(149, 0.3790370241069617, 185)	0.01175	0.08804	0.30913
	2	(104, 0.01, 252)	0.01012	0.08273	0.28536
	3	(142, 0.01, 300)	0.00928	0.07814	0.26950
	4	(133, 0.01, 300)	0.00894	0.07789	0.26171
	5	(133, 0.01, 300)	0.00894	0.07789	0.26171
5	1	(217,	0.00893	0.07929	0.27346

Trial	Iteration	Configuration	MSE	MAE	MAPE
		0.043700556915905936, 132)			
	2	(245, 0.06711297795558117, 125)	0.00875	0.07657	0.27046
	3	(245, 0.06711297795558117, 125)	0.00875	0.07657	0.27046
	4	(245, 0.06711297795558117, 125)	0.00875	0.07657	0.27046
	5	(245, 0.06711297795558117, 125)	0.00875	0.07657	0.27046

The testing results on LSTM-MBES-AdaBoost model have shown improvements in key performance metrics, namely MSE, MAE, and MAPE when compared to previous model variations such as LSTM, LSTM-BES, and LSTM-MBES. Specifically, the LSTM-MBES-AdaBoost managed to obtain lower MSE, MAE, and MAPE. In addition, in trial 5, using the solution parameters (245, 0.06711297795558117, 125) yielded the least errors. This particular solution outperformed other hyperparameters obtained from LSTM, LSTM-BES, and LSTM-MBES model; thus, demonstrating the efficiency in minimizing prediction errors.

6. Comparison of All Models

Table 9 shows the results of overview comparison performance between all models. The MSE, MAE, and MAPE values are obtained and summarized from the best results in each of the trial for each of the method. In Table 9, it shows the best result of LSTM model can be found in trial 2. It has MSE = 0.01024, MAE = 0.08131, and MAPE =0.25930. The configurations of parameters (units in layer) of LSTM model are performed through trials and errors. The results obtained in LSTM are merely served as a benchmark to evaluate the performance and compare with three other models. Whereas, optimization is being performed on LSTM-BES, LSTM-MBES, and LSTM-MBES-AdaBoost to obtain the best solution of parameters to fit into the model.

Table 9. Overview Comparison of All Models

Method	Trial	MSE	MAE	MAPE
LSTM	1	0.01379	0.09416	0.29377
	2	0.01024	0.08131	0.25930
	3	0.01198	0.08788	0.31779
	4	0.01222	0.08921	0.30949
	5	0.01052	0.08139	0.25738
	Average	0.01175	0.08679	0.28755
LSTM-BES	1	0.01169	0.08766	0.29988
	2	0.01116	0.08416	0.28185
	3	0.01119	0.08738	0.33416
	4	0.01120	0.08459	0.31663
	5	0.01121	0.08299	0.26692
	Average	0.01129	0.08536	0.29988
LSTM-MBES	1	0.01001	0.08008	0.27867
	2	0.01069	0.08332	0.27936
	3	0.01121	0.08457	0.27186
	4	0.01153	0.08569	0.28697
	5	0.01162	0.08476	0.26253
	Average	0.01101	0.08368	0.27588
LSTM-	1	0.00963	0.08042	0.29122

Method	Trial	MSE	MAE	MAPE
MBES-AdaBoost	2	0.00895	0.07749	0.26698
	3	0.00912	0.0795	0.28318
	4	0.00894	0.07789	0.26171
	5	0.00875	0.07657	0.27046
	Average	0.00908	0.07837	0.27471

It can be seen that the performance of LSTM-MBES is slightly better than LSTM-BES. In LSTM-MBES, the best result can be found in trial 1; having MSE = 0.01001, MAE = 0.08008, and MAPE = 0.27867. The trial 1 result in LSTM-MBES is the lowest MSE and MAE values compared to the other trials. While in LSTM-BES, the best result is in trial 2 with the lowest MSE; where the MSE = 0.01116, MAE = 0.08416, and MAPE = 0.28185. Observing from the differences between LSTM-BES and LSTM-MBES, MSE has different by 0.00115 (approximately reduced by 10.30%), MAE has different by 0.00408 (approximately reduced by 4.84%), and MAPE has different by 0.00318 (approximately reduced by 1.13%).

Comparing the LSTM-MBES with LSTM-MBES-AdaBoost, overall, the LSTM-MBES-AdaBoost has shown significant good results. Since both LSTM-MBES and LSTM-MBES-AdaBoost are using the same configuration of parameters (units in layers) which are obtained from the best solution in the optimization results, thus, it can be evaluated side by side. The best performance of LSTM-MBES-AdaBoost can be found in trial 5; where it has the lowest MAE and MAPE values among other trials. It has MSE = 0.00875, MAE = 0.07657, and MAPE = 0.27046. Comparing both trial 5 results in LSTM-MBES and LSTM-MBES-AdaBoost (since both are using same configuration of parameters), it can be observed that, after performed LSTM-MBES-AdaBoost; the MSE has different by 0.00287 (approximately reduced by 24.69%), MAE has different by 0.00819 (approximately reduced by 9.66%), and MAPE has different by 0.00793 (approximately increased by 2.93%).

Despite in LSTM-MBES trial 5 result in the MAPE, it has lower MAPE value compared to LSTM-MBES-AdaBoost trial 5; however, overall, the LSTM-MBES-AdaBoost has improved the MSE by 24.69% and MAE by 9.66%. In addition, looking from the average values from all of the trials in all of the models; it can be concluded that LSTM-MBES-AdaBoost has the lowest average value where MSE = 0.00908, MAE = 0.07837, and MAPE = 0.27471. This indicates that LSTM-MBES-AdaBoost has the lowest error rate compared to other models and has the best performance among other models.

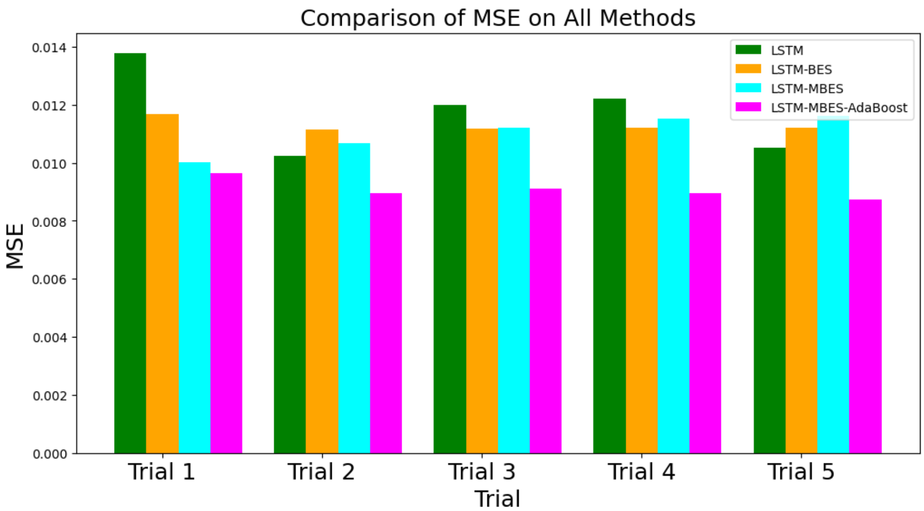

Figure 6. Comparison of MSE on All Methods

Figure 6 shows the graphs of comparison of MSE between LSTM, LSTM-BES, LSTM-MBES, and LSTM-MBES-AdaBoost. In the Figure 6, it can be observed that LSTM-MBES-AdaBoost has the lowest MSE values in all of the trials compared to other methods. This indicates that LSTM-MBES-AdaBoost has the best performance among the others. LSTM-MBES managed to outperform LSTM-BES in trial 1 and trial 2. However, LSTM-BES is slightly better in trial 4 and trial 5 when compared to LSTM-MBES. The highest MSE values is founded in trial 1 of LSTM.

Figure 7 shows the graphs of comparison of MAE between LSTM, LSTM-BES, LSTM-MBES, and LSTM-MBES-AdaBoost. It can be observed that LSTM-MBES-AdaBoost has the lowest MAE values in all of the trials compared to other methods. This indicates that LSTM-MBES-AdaBoost has the best performance among other models. In addition, LSTM-MBES managed to outperform LSTM-BES as it has lower MAE values in trial 1, trial 2, and trial 3. The highest MAE value is located at trial 1 of LSTM.

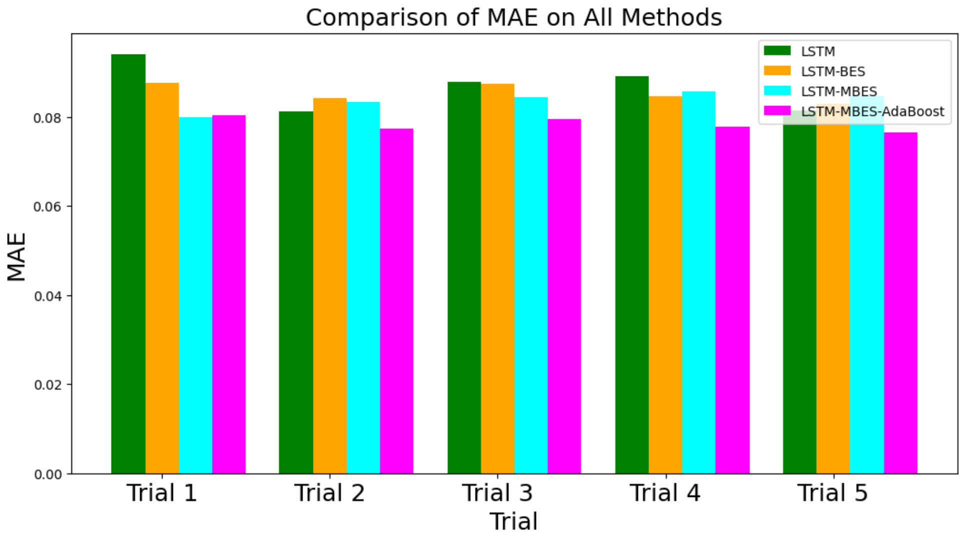

Figure 7. Comparison of MAE on All Methods

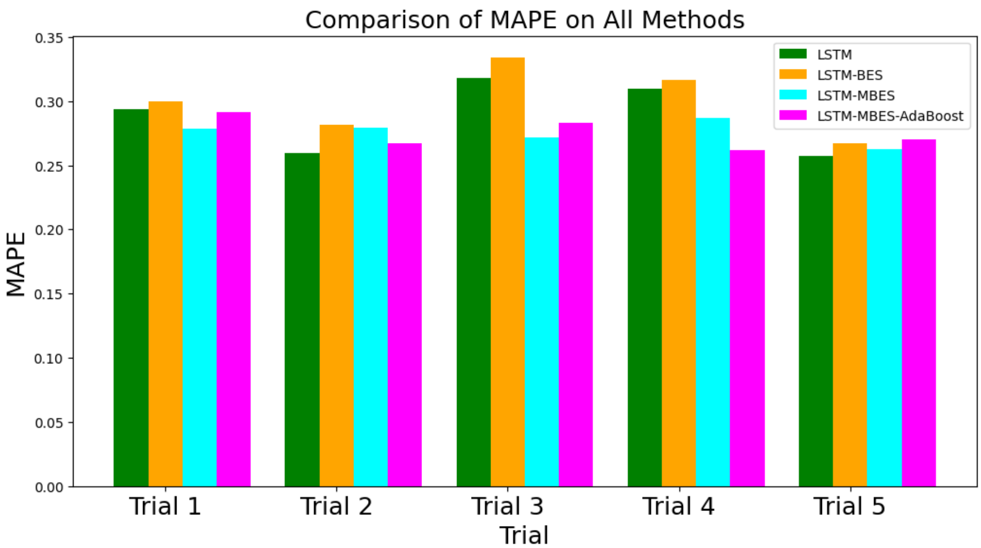

Figure 8. Comparison of MAPE on All Methods

Figure 8 shows the graphs of comparison of MAPE between LSTM, LSTM-BES, LSTM-MBES, and LSTM-MBES-AdaBoost. In the Figure 10, it can be observed that highest MAPE value is located in trial 3 of LSTM-BES. Overall, the LSTM-BES has higher MAPE values in all of the trials compared to LSTM-MBES and LSTM-MBES-AdaBoost. While LSTM is configured through trials and errors, it has slightly better performance in trial 2 and trial 5, however, worst performance is in trial 3 and trial 4.

Table 10 shows the initial best solution cost between LSTM-BES and LSTM-MBES or LSTM-MBES-AdaBoost specifically.

Table 10. Overview Comparison of Initial Best Solution Cost

Trial	Iteration	Cost	
		LSTM-BES	LSTM-MBES or LSTM-MBES-AdaBoost
1	1	0.01473	0.01279
	2	0.01473	0.01028
	3	0.01473	0.01028
	4	0.01289	0.01028
	5	0.01289	0.01028
	Margin (between 1st and 5th iteration)	12.44 %	19.62%
2	1	0.01207	0.01256
	2	0.01207	0.01256
	3	0.01207	0.01256
	4	0.01207	0.01256
	5	0.01207	0.01256
	Margin (between 1st and 5th iteration)	0%	0%
3	1	0.01482	0.01373
	2	0.01302	0.01282
	3	0.01302	0.01246
	4	0.01302	0.01246
	5	0.01302	0.01246
	Margin (between 1st and 5th iteration)	12.12%	9.21%
4	1	0.01455	0.01358
	2	0.01455	0.01358
	3	0.01455	0.01328
	4	0.01455	0.01283
	5	0.01224	0.01244
	Margin (between 1st and 5th iteration)	15.89%	8.45%
5	1	0.01381	0.01306
	2	0.01381	0.01306
	3	0.01249	0.01202
	4	0.01240	0.01202
	5	0.01240	0.01202
	Margin (between 1st and 5th iteration)	10.19%	7.93%

In Table 10, it can be seen that almost in all of the trials in the 1st iteration, the LSTM-MBES or LSTM-MBES-AdaBoost has lower cost values compared to LSTM-BES. This has shown that LSTM-MBES or LSTM-MBES-AdaBoost has better performance in determine initial best solution. Observing from the margin difference between 1st and 5th iteration, the best performance of LSTM-MBES or LSTM-MBES-AdaBoost can be identified in trial 1 where it managed to reduce the fitness cost value by 19.62%.

In some trials, it might seem that LSTM-BES has higher and better margin differences compared to LSTM-MBES or LSTM-MBES-AdaBoost such as in trial 3, trial 4, and trial 5. However, in the 5th iteration of trial 1, trial 2, and trial 5; LSTM-MBES/LSTM-MBES-AdaBoost managed to obtain lower cost compared to LSTM-BES. This indicates that LSTM-MBES/LSTM-MBES-AdaBoost able to find more optimize solution as the final cost is lower compared to LSTM-BES.

7. Conclusions and Future Works

This paper proposed a hybrid of LSTM-MBES-AdaBoost model to further optimize and improve the capabilities of LSTM through combining with Modified Bald Eagle Search, Quasi opposition-based learning, and AdaBoost approach. The experimental results have shown that the proposed method has shown better results as it is able to obtain least MSE, MAE, and MAPE compared to the LSTM, LSTM-BES, and LSTM-MBES models.

In addition, the proposed method has better performance in obtaining the initial best solution during 1st iteration as shown in Table 10. The proposed method also improved the initial best solution cost during the iteration process which is shown in the optimization results presented in Table 6. The proposed method, LSTM-MBES-AdaBoost, utilizes the same parameter configurations as LSTM-MBES when feeding parameters into LSTM model for testing. The best performance is LSTM-MBES-AdaBoost managed to improve the cost by 19.62% in one of the trials. However, in other trials, LSTM-BES able to improve the cost with higher margin compared to LSTM-MBES-AdaBoost such as trial 3, trial 4, and trial 5 as shown in Table 10. Nonetheless, LSTM-MBES-AdaBoost outperformed LSTM-BES in obtaining the lowest final cost which happened in 3 out of 5 trials in the 5th iteration (as shown in Table 10).

As a future work, since some of the parameters in Bald Eagle Search are still predefined such as A_factor, R_factor, c1, and c2 values; this study can be extended to further optimize those values so that it can be determined automatically for the most optimum parameters. In addition, further enhancement on select stage, search stage, and swoop stage of Bald Eagle Search can be implemented to improve the accuracy better. Also, performance testing on the algorithm can be explored using other concrete compressive strength dataset from various companies or sources because each company might use different components when mixing the concrete. Therefore, the forecasting accuracy of the model can be further improved in the future.

8. References

[1]. Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, *53*(8), 5929-5955

[2]. Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research*, *270*(2), 654-669.

[3]. Cen, Z., & Wang, J. (2019). Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer. *Energy*, *169*, 160-171.

[4]. Li, X., Zhang, L., Wang, Z., & Dong, P. (2019). Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and Elman neural networks. *Journal of Energy Storage*, *21*, 510-518.

[5]. Wu, Y. X., Wu, Q. B., & Zhu, J. Q. (2019). Improved EEMD-based crude oil price forecasting using LSTM networks. *Physica A: Statistical Mechanics and its Applications*, *516*, 114-124.
[6]. Xiao, C., Chen, N., Hu, C., Wang, K., Gong, J., & Chen, Z. (2019). Short and mid-term sea surface temperature prediction using time-series satellite data and LSTM-AdaBoost combination approach. *Remote Sensing of Environment*, *233*, 111358.
[7]. Sun, S., Wei, Y., & Wang, S. (2018, June). AdaBoost-LSTM ensemble learning for financial time series forecasting. In *International Conference on Computational Science* (pp. 590-597). Springer, Cham.
[8]. Ting, T. O., Yang, X. S., Cheng, S., & Huang, K. (2015). Hybrid metaheuristic algorithms: past, present, and future. *Recent advances in swarm intelligence and evolutionary computation*, 71-83.
[9]. Tuerxun, W., Xu, C., Guo, H., Guo, L., Zeng, N., & Gao, Y. (2022). A Wind Power Forecasting Model Using LSTM Optimized by the Modified Bald Eagle Search Algorithm. *Energies*, *15*(6), 2031.
[10]. Wu, J., Hu, Y., Wu, D., & Yang, Z. (2022). An Aquatic Product Price Forecast Model Using VMD-IBES-LSTM Hybrid Approach. *Agriculture*, *12*(8), 1185.
[11]. UCI Machine Learning Repository. Concrete Compressive Strength Data Set. Retrieved from 30th October 2022. https://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength
[12]. Nikoo, M., Torabian Moghadam, F., & Sadowski, Ł. (2015). Prediction of concrete compressive strength by evolutionary artificial neural networks. *Advances in Materials Science and Engineering*, *2015*.
[13]. Chen, H., Li, X., Wu, Y., Zuo, L., Lu, M., & Zhou, Y. (2022). Compressive strength prediction of high-strength concrete using long short-term memory and machine learning algorithms. *Buildings*, *12*(3), 302.
[14]. Khademi, F., Akbari, M., Jamal, S. M., & Nikoo, M. (2017). Multiple linear regression, artificial neural network, and fuzzy logic prediction of 28 days compressive strength of concrete. *Frontiers of Structural and Civil Engineering*, *11*(1), 90-99.
[15]. Zheng, T., & Luo, W. (2019). An enhanced lightning attachment procedure optimization with quasi-opposition-based learning and dimensional search strategies. *Computational Intelligence and Neuroscience*, *2019*.
[16]. Chen, H., Li, W., & Yang, X. (2020). A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems. *Expert Systems with Applications*, *158*, 113612.
[17]. Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), 119-139.
[18]. Wang, R. (2012). AdaBoost for feature selection, classification and its relation with SVM, a review. *Physics Procedia*, *25*, 800-807.
[19]. Kumar, A. (2022, January 9). *The Ultimate Guide to AdaBoost Algorithm | What is AdaBoost Algorithm?*. Great Learning. Retrieved from 27th September 2022. https://www.mygreatlearning.com/blog/adaboost-algorithm/
[20]. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.
[21]. Tanyildizi, H., Şengür, A., Akbulut, Y., & Şahin, M. (2020). Deep learning model for estimating the mechanical properties of concrete containing silica fume exposed to high temperatures. Frontiers of Structural and Civil Engineering, 14(6), 1316-1330.
[22]. Achkar, R., Elias-Sleiman, F., Ezzidine, H., & Haidar, N. (2018). Comparison of BPA-MLP and LSTM-RNN for Stocks Prediction. *2018 6th International Symposium on Computational and Business Intelligence (ISCBI)*, 48-51.
[23]. Comprehensive guide to LSTM & RNNs. Retrieved from 10th January 2023. https://www.turing.com/kb/comprehensive-guide-to-lstm-rnn

[24]. Alsattar, H. A., Zaidan, A. A., & Zaidan, B. B. (2020). Novel meta-heuristic bald eagle search optimisation algorithm. *Artificial Intelligence Review*, *53*(3), 2237-2264.
[25]. Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. (2007, September). Quasi-oppositional differential evolution. In *2007 IEEE congress on evolutionary computation* (pp. 2229-2236). IEEE.
[26]. Wang, J. (2022, July). Prediction of Concrete Compressive Strength Based on Improved EM Algorithm and Ridge Regression Algorithm. In *2022 International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS)* (pp. 59-62). IEEE.
[27]. Yunani, J. L. (2019, October). Concrete compressive strength analysis using tree-based methods. In *2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)* (pp. 1-4). IEEE.

Herny Christine received her Bachelor of Science in Computing from President University in 2021. She is currently pursuing Master of Science in Information Technology degree at President University. Her research interests are machine learning and blockchain.

Rila Mandala received his Bachelor degree in Information Technology from ITB, Bandung in 1992. He completed his Master of Engineering in Computer Science and Doctor of Philosophy (Ph.D) in Computer Science, both from Tokyo Institute of Technology, Japan in 1998 and 2001 respectively. Since 1992, he has been a permanent lecturer in the School of Electrical Engineering and Informatics at ITB, Bandung for undergraduate and postgraduate study programs.