Received January 20, 2021, accepted April 6, 2021, date of publication April 19, 2021, date of current version April 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3073876

An Improved Alpha Beta Filter Using a Deep Extreme Learning Machine

JUNAID KHAN[1], MUHAMMAD FAYAZ[2], AYYAZ HUSSAIN[3], (Member, IEEE), SHAH KHALID[4], WALI KHAN MASHWANI[5], AND JEONGHWAN GWAK[6,7,8,9], (Member, IEEE)

[1]College of Information and Communication Engineering, Intelligent Systems Research Institute, Sungkyunkwan University, Suwon 440-746, South Korea
[2]Department of Computer Science, University of Central Asia, Naryn 722918, Kyrgyzstan
[3]Department of Computer Science, Quaid-i-Azam University, Islamabad 45320, Pakistan
[4]Department of Computer Science and IT, University of Malakand, Chakdara 18800, Pakistan
[5]Institute of Numerical Sciences, Kohat University of Science and Technology, Kohat 26000, Pakistan
[6]Department of Software, Korea National University of Transportation, Chungju 27469, South Korea
[7]Department of Biomedical Engineering, Korea National University of Transportation, Chungju 27469, South Korea
[8]Department of AI Robotics Engineering, Korea National University of Transportation, Chungju 27469, South Korea
[9]Department of IT and Energy Convergence (BK21 FOUR), Korea National University of Transportation, Chungju 27469, South Korea

Corresponding author: Jeonghwan Gwak (jgwak@ut.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2020R1I1A3074141, in part by the Brain Research Program through the NRF funded by the Ministry of Science, ICT and Future Planning under Grant NRF-2019M3C7A1020406, and in part by the Regional Innovation Strategy (RIS) through the NRF funded by the Ministry of Education.

ABSTRACT This paper introduces new learning to the prediction model to enhance the prediction algorithms' performance in dynamic circumstances. We have proposed a novel technique based on the alpha-beta filter and deep extreme learning machine (DELM) algorithm named as learning to alpha-beta filter. The proposed method has two main components, namely the prediction unit and the learning unit. We have used the alpha-beta filter in the prediction unit, and the learning unit uses a DELM. The main problem with the conventional alpha-beta filter is that the values are generally selected via the trial-and-error technique. Once the alpha-beta values are chosen for a specific problem, they remain fixed for the entire data. It has been observed that different alpha-beta values for the same problem give different results. Hence it is essential to tune the alpha-beta values according to their historical behavior for certain values. Therefore, in the proposed method, we have addressed this problem and added the learning module to the conventional α-β filter to improve the α-β filter's performance. The DELM algorithm has been used to enhance the conventional alpha-beta filter algorithm's performance in dynamically changing conditions. The model performance has been measured using indoor environmental values of temperature and humidity. The relative improvement in the proposed learning prediction model's accuracy was 7.72% and 16.47% in RMSE and MSE metrics. The results show that the proposed model outperforms in terms of the result as compared to the conventional alpha-beta filter.

INDEX TERMS Alpha-beta filter, learning algorithm, prediction algorithm, deep extreme learning machine, energy prediction.

I. INTRODUCTION

It is essential to predict accurate and reliable parameters for the energy consumption prediction algorithms. All decision-making systems need a perfect conception of upcoming trends and risks to predict future results because past results may not be enough to make the ideal decision. Hence, it is essential to estimate future values accurately because any wrong estimation can lead to unforeseen consequences and performance decreases in the prediction algorithms [1]. The simple way to predict the future values is to analyze the data, acquire some results, and based on results, predict future values. This kind of practice requires a lot of training time; hence an intelligent system is always needed to make correct decisions in a short time.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhan-Li Sun.

There is a requirement to pre-process the data in every energy prediction model before providing it as input to prediction algorithms [2]. Different techniques have been used to pre-process data for energy consumption prediction, like the Kalman filter, Alpha-Beta filter.

Assigning appropriate values to alpha-beta parameters in the alpha-beta filter is challenging because once the values have been assigned to these parameters, they remain fixed for the whole data. However, different alpha-beta values provide different results; hence it is necessary to tune the alpha-beta values according to their historical behavior for certain values to achieve better energy prediction results.

The systems based on machine learning approaches, optimization algorithms, and mathematical models help energy consumption prediction and environmental parameter prediction [3]. The most significant and rapidly expanded research area is machine learning, wherein many applications have been initiated and developed for prediction purposes in different fields. Machine Learning algorithms can be categorized as supervised, unsupervised, semi-supervised, reinforcement learning, and so forth [4]. Supervised learning needs supervision in the form of labeled data for learning the prediction model. The learning model connects input and output parameters, and this developed connection is later used for input to predict the output. Some popular and common supervised learning algorithms are support vector machine [5], artificial neural networks [6], classification and regression tree [7], and so forth. These supervised learning algorithms require historical data for the training, and then the trained models are used for prediction in specific areas [8]. The problem related to this type of algorithm is that they are less effective in dynamic circumstances or do not produce effective results like rapidly changing indoor environments in smart homes.

Many ensemble algorithms, such as mixture-of-experts, stacked generalization, have been designed for the multi-purpose to predict and classify [9]. The ensemble methods perform better than simple machine learning methods; for instance, stacked generalization performs better than single-neural network approaches [10]. Another ensemble method is mixture-of-experts, which uses different statistical estimations for performance enhancement. Due to statistical estimation, the mixture-of-experts approach's prediction-accuracy is high compared to the other methods [9].

The subset of machine learning is Deep Learning. In Artificial Intelligence, the deep learning network also has the competency of learning with unsupervised statistics that is unlabeled, unstructured, or semi-structured DL, also known as deep neural network or deep neural learning [11]. Because of the working and its function like the human brain, DL can process the data and create patterns for the usage of the decision-making systems [12].

Deep Extreme Learning Machine is another most popular method, which is the mixture of deep learning (DL) and an Extreme Learning Machine algorithm that has also been used in energy consumption prediction and different other fields for the prediction and classification problems [13]. The ordinary artificial neural network (ANN) based algorithm requires larger data samples for training, higher computation time, and their training is slower for learning. Further, sometimes the learning model may lead to over-fitting due to extra training [13]. Hence, the DELM has all the capabilities to be used for the tuning of parameters in the alpha-beta filter.

Assigning appropriate values to alpha-beta parameters in the alpha-beta filter is a challenging task because once the values have been assigned to these parameters, they remain fixed for the whole data. But different alpha-beta values provide different results; hence it is necessary to tune the alpha-beta values according to their historical behaviour for certain values to achieve better results [13], [14].

The selection of the input weights and hidden layers, the biases have a significant impact on the performance of the ELM module [15], [16], and it has been proved that their initialization methods are related to the statistical characteristics of the input data [17]. The extreme learning machine approaches can be used for classification, clustering, regression, feature learning, compression, and sparse function approximation and pattern classification with either a single layer or multilayer of hidden nodes [18].

DELM is another most popular method, which is the mixture of deep learning (DL) and an ELM algorithm that has also been used in energy consumption prediction and different other fields for the prediction and classification problems [18].

The ordinary artificial neural network (ANN) based algorithm requires larger data samples for training, higher computation time, and their training is slower for learning. Further, sometimes the learning model may lead to over-fitting due to extra training [18]. Hence, the DELM has all the capabilities to be used to the tuning of parameters in the alpha-beta filter. DELM can help and be used for both classification and regression problems [19].

This paper has proposed a new methodology for tuning the alpha-beta filter parameters based upon a deep extreme learning machine and named it as learning to alpha-beta filter. The proposed method has two main components, namely the prediction unit and the learning unit. We have used the deep extreme learning machine (DELM) in the prediction unit, and the second unit uses the α-β filter [20]. Therefore, in the proposed method, we have addressed the problem of static parameters and added a learning module to the conventional alpha-beta filter to tackle this problem. For the performance analysis, we have applied the proposed learning to the alpha-beta filter on real data of temperature and humidity and compared the results with the conventional alpha-beta filter data to check effectiveness. The learning module has improved the overall performance of the α-β filter algorithm. The proposed method output will be helpful for the improvement of the performance of energy prediction algorithms. Table 1 shows different types of symbols and notations with descriptions used in our paper.

The rest of the paper is organized as; the related work is briefly discussed in section II. The proposed methodology

TABLE 1. The list of symbols and notations used in this paper.

Symbol	Description	Symbol	Description	Symbol	Description
α-β	Alpha Beta	$x_{k-1} = c_1$	Initialization of the alpha-beta values	$H_1 = V\gamma^+$	Output neuron in the 2nd hidden layer
P	Input layer node	$v_{k-1} = c_2$	Initialization of the alpha-beta values	γ_{new}	Weight matrix
Q	Hidden layer node	x_k	Update the position and predicting position	$[b_{1,} b_{2,} b_{3,}$	Biases and its Transpose
r	Output layer node	x_m	Reading Sensor data	h_{cur}	Current humidity values
[A, B]	Training samples	Δx_k	Computing the difference	h_{min}	Minimum humidity values
a and b	Input and output matrix features	v_k	Predicted velocity	h_{max}	Maximum humidity values
W	Weights	A	Actual values	Tanh	hyperbolic tangent function
N	Total number of observations	Err	Error	RMSE	Root Mean Square Error
V	Resultant matrix	P	Predicted values.	MAE	Mean Absolute Error
V'	Transpose of the resultant matrix	CC	Correlation coefficient	MSE	Mean Square Error

is comprised of different subsections and presented in section III. Section IV shows a brief discussion on Experimental analysis and results. The discussion is provided in section V. The conclusion of the paper is presented in section VI.

II. RELATED WORKS

With the growth of the storage and processing power of computers, a tremendous increase in the use of machine learning (ML) algorithms and the introduction of new algorithms for solving different problems have been reported. Various authors have used machine learning algorithms, ANN, and deep learning for different purposes. In this section, some of the prominent uses of these algorithms in different fields have been reported [1], [8], [11].

Kang *et al.* also suggested a method based upon fuzzy inference for performance tuning of the K filter [21]. Ibarra *et al.* suggested a scheme based upon gyroscope and acceleration sensors called adaptive neuro-fuzzy-inference for the parameters tuning of K filter for accurate attitude approximation [22]. For this method, the Markov model has also been used [23].

The Kalman filter has been modified for the linearization of the estimation of the existing mean and covariance. The iterated E-Kalman filter is the modification of the original filter to calculate the state estimates as a maximum posterior. The Gauss-Newton method [24] has been used to update the parameters. The basic Kalman filter and extended Kalman filter are suitable for a smaller number of parameters. Ensembled Kalman filter is suitable for the situation where the processing of a large number of parameters is required. The particle filter is similar in characteristics with the Kalman filter.

Still, the main difference lies between the Kalman filter and the particle filter is that the Kalman filter makes the supposition that all the probability involved are Gaussian [25], [26]. The Kalman filter's integration with other techniques also has been carried out, i.e., with an unbiased finite-impulse response (UFIR) filter. These filters provide robustness but degrade the accuracy of the algorithm [27]. The approach based on machine learning and decision-making system is an adaptive neural network [28].

The improved fuzzy alpha-beta filter has been proposed in [29] to track the highly maneuvering targets. A simplified alpha-beta filter based on Gaussian sum has been used by [30]. An EP-based α–β–γ–δ filter has been used in [31] for the target tracking.

Jamil *et al.* proposed a new methodology for accuracy enhancement of α-β filter with ANN-based learning tool in the internal navigation system. The development of small sensors makes it easy to track the entity in an interior room atmosphere. These sensors can deal with information about the user like patterns, communication, posture, location, which helps catch the user perspectives. Interior Localization is a new challenging area and getting a lot of consideration, while attention towards position-based facilities is also growing. Jamil *et al.* proposed a new methodology for the accuracy improvement of interior direction-finding using a learning-to-prediction ideal based on ANN. In an indoor environment GPS not working properly, the proposed system tack the position of an individual in an interior location. The results show that the proposed filter with a learning component outperforms compared to the outdated α-β filter. The main drawback is that the algorithm tested for the fixed value of alpha and beta [32].

So these filters have been used in different fields to solve the problems. The machine learning algorithms also have importance and have been used in many areas, as Yates *et al.* [33] Presented a new technique based upon the K-nearest neighbor for generating alternative climate dataset conditions. In [34], a new technique, K-nearest neighbor (KNN) multi-label multi-instance learning, has been applied for text categorization and scene classification. Gunn in [6] used the SVM for regression and classification. Suthaharan in [35] discussed the ML algorithms for the classification of big data analysis.

The classification and regression tree have also been used in different areas for classification and regression problems. The CART problem has the potential to resolve the complex interrelationship between predictive factors. The CART algorithm merges two trees: classification and regression tree, constructed for the dependent variable with order and without order correspondingly. The square difference is used to compute the inaccuracy between predicted values and actual values in the data [36].

One of the most popular prediction algorithms and widely used for different areas is ANN, which is not application-specific; therefore, it can solve numerous problems. The ANN can resolve the problems of classification, regression, clustering, and pattern recognition problems. If the problem is classification type, then the number of the output layer and the number of neurons in the hidden layer relies on the number of classes. While in the case of regression, only a single neuron is specified in the output layer [6], [7], [33], and [37].

Nowadays, deep learning methods are also used for classification and regression purposes [19]. The numbers of hidden layers in deep learning are more than one as opposed to a simple neural network. The deep learning methods perform better when the input parameters are greater in number. Some famous deep learning algorithms are convolutional neural networks, deep extreme learning machines, [11], [38]. The ANN has been combined with many other techniques, such as the Sugeno fuzzy inference technique [39] and Mamdani fuzzy inference technique [40]. The Sugeno fuzzy inference system and the feedforward neural networks both combine and make a new method named adaptive-neuro-fuzzy inference system (ANFIS) [41]. This system has the capability of mapping human intelligence in the shape of rules. The ANFIS has five layers; the input data are used as parameters to the input layer, which is the first layer of ANFIS. The fuzzy system's membership function is determined for the input layer's respective input neurons in the second layer; the rules are defined in the third layer of the ANFIS. The membership function is defined in the fourth layer for output neurons. The last layer generates the output of the ANFIS. Different authors have used the ANFIS for various purposes, such as Fayaz and Kim [11] used the ANFIS for energy consumption prediction in the smart home. Wang *et al.* [42] used a technique based on ANFIS for risk assessment of the bridge. Kassa *et al.* [43] worked on another technique for short-term wind power prediction based on ANFIS.

Fayaz *et al.* [44] Suggested a new technique based upon Mamdani fuzzy-logic and feedforward backpropagation NN for the consumption of energy in smart electric vehicles. This method consists of a learning module and a control module. The control module used the Mamdani fuzzy logic system, and in the learning module, they also used the feedforward backpropagation neural network (FFBNN). The learning module's output is the desired membership functions for the fuzzy logic controller, and the mandatory power for actuators is the output of the fuzzy logic controller.

Fayaz and Kim [11] suggested a novel approach based upon the DELM for the consumption of energy in the residential building. This method has improved the performance and energy consumption prediction accuracy. Ding *et al.* [45] also proposed a new approach that focused on applying the deep ELM for the experimental analysis of the visual-feedback and classification of brain-computer interface (BCI) for EEG classification using a deep extreme learning machine. The DELM result was very useful in EEG classification when combined MLELM and ELM with kernel and put forward to DELM. Hoang and Bui [46] carried out a relative method for the prediction of slope steadiness using different types of innovative machine learning approaches and computing algorithms using extreme learning machines, Least Squares SVM (LSSVM), and Radial Basis Function Neural Network (RBFNN). The binary classification has been used for slope assessment. The research indicates that LSSVM and ELM are both superior in terms of performance to RBFNN.

Huang [47] suggested a new approach to compare extreme learning machines with feedforward NNs. The study indicates that the learning rate of the ELM is too much higher than the feedforward neural network (FFNN). The study also suggested two reasons behind the slow learning rate of FFNN that are slow gradient-based learning, and all the parameter values of the model are tuned recursively.

III. METHODOLOGY

All the algorithms used for prediction are usually trained by using historical data to achieve better prediction accuracy. The training lets the algorithms learn the data's hidden patterns and the relationship between input and output parameters. The models that have been trained are used to take input data or parameters and predict the output values. The prediction algorithm will outperform when the application scenario and the given input data remain constant as the training data circumstances. In contrast, all the available prediction algorithms do not permit modification on the trained model to deal with dynamic scenarios. The proposed learning to prediction model overcomes this limitation and modifies the alpha-beta filter parameters according to dynamic conditions.

The conceptual model of our proposed algorithm is presented in Figure 1.

The proposed model comprises two main modules, namely, prediction and learning modules. In the prediction module, inputs are different parameters (sensor temperature and sensor humidity), and its output is the predicted parameter values (predicted temperature and predicted humidity). The learning module inputs are the parameters that are needed to be predicted. It also uses other parameters because of the impact on

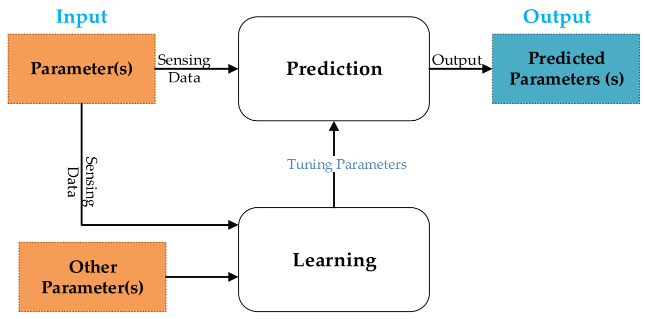


FIGURE 1. Conceptual learning to the prediction model.

parameters that need to be predicted. The learning module is used to tune some parameter values in the prediction module. The detailed working mechanisms of learning to alpha-beta filter are shown in Figure 2.

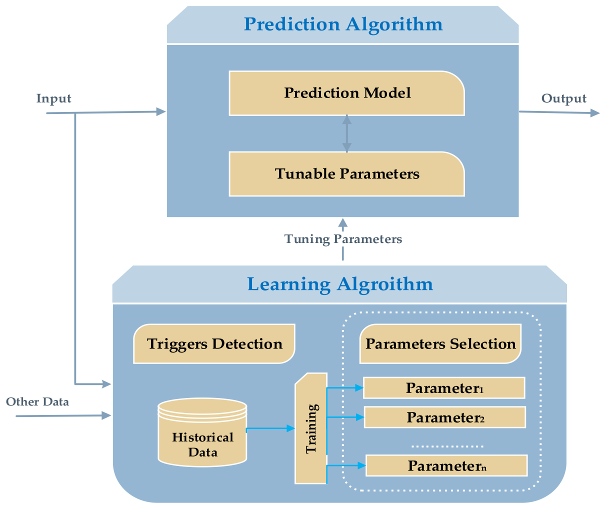


FIGURE 2. The working mechanism of the alpha-beta filter.

For experimental analysis of the algorithm, the alpha-beta filter as the prediction algorithm has been used to predict the parameters. The proposed learning component is based upon the DELM, as can be seen in Figure 3. The α-β filter does not need the entire historical data information. However, only the preceding state data makes the algorithm more and more intelligent for prediction and determination of the actual state of the system, which is because it is a lightweight algorithm [31]. In this study, we have used this filter to predict the real temperature values from the noisy readings in temperature sensor data, whereas noisy readings in the temperature sensor depend upon the scenario under which the readings in the temperature sensor are deeply affected by the level of humidity in the surroundings. In the conventional alpha-beta filter, the values are static. The fixation of values is the main weakness of this algorithm because different alpha-beta filter provides different results on different alpha-beta filter values. So, the primary aim of this research contribution is to tune the alpha-beta values based on some previous experience for every input. In our proposed model, we have used the learning module comprised of DELM for tuning the alpha-beta values to increase the performance.

In the learning unit of the model, the DELM network takes two parameters as an input such as the values of current temperature and the level of humidity. The alpha-beta filter acquires data from the temperature sensor with respect to time, and guess the real temperature Tt by the elimination of noise. The performance of our filter is primarily measured and managed by tunable parameters identified by alpha and beta. After every iteration, these tunable parameters are updated. In the learning module, the DELM based on tunable parameters will find out the projected error in readings sensors, so that alpha and beta values can be updated intelligently.

The following subsection describes the DELM and the alpha-beta filter in detail.

A. DEEP EXTREME LEARNING MACHINE (DELM)

DELM is the most popular method, which is the mixture of deep learning (DL) and an ELM algorithm that has also been used in energy consumption prediction and different other fields for the prediction and classification problems [13]. The ordinary ANN-based algorithm requires more data samples for training, more computation time, and slower training for learning, and sometimes the learning model may lead to over-fitting [18]. The ELM learns very quickly and computationally efficiently, and because of its efficiency, it is widely used in different areas, such as classification and regression purposes [48]. This model is comprised of two input layers, different hidden layers, and a single output layer. The ELM structure diagram is depicted in Figure 4, whereas the input layer nodes are represented by p, hidden layer nodes are denoted by q, and output layer nodes are indicated by r.

Initially, we take training samples $[A, B] = \{a_{k,} b_{k,}\}(i = 1, 2, \ldots, Z)$, input feature $A = [a_{k1}a_{k2}a_{k3} \ldots\ldots a_{kZ}]$ and a targeted matrix $B = [b_{l1}b_{l2}b_{l3} \ldots\ldots b_{lZ}]$. It comprises samples for training, matrices A, B that can be represented as in (1) and (2), respectively. The input and output matrix features are represented by a and $b,$ and then ELM arbitrarily fine-tuned the weights between the input and the hidden layers. While w_{kl} are the weights between the Nth input layer nodes and Ith hidden layer nodes, as depicted in (3). The hidden layer parameters of ELM are randomly generated, and the weights between hidden and output layer neurons are calculated through a least-square manner, which is represented in (4), and the weights between the input and hidden layers nodes are represented by γ_{kl}.

$$\mathrm{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1z} \\ a_{21} & a_{22} & \cdots & a_{2z} \\ a_{31} & a_{32} & \cdots & a_{3z} \\ \vdots & \vdots & & \vdots \\ a_{p1} & a_{p2} & \cdots & a_{pz} \end{bmatrix} \tag{1}$$

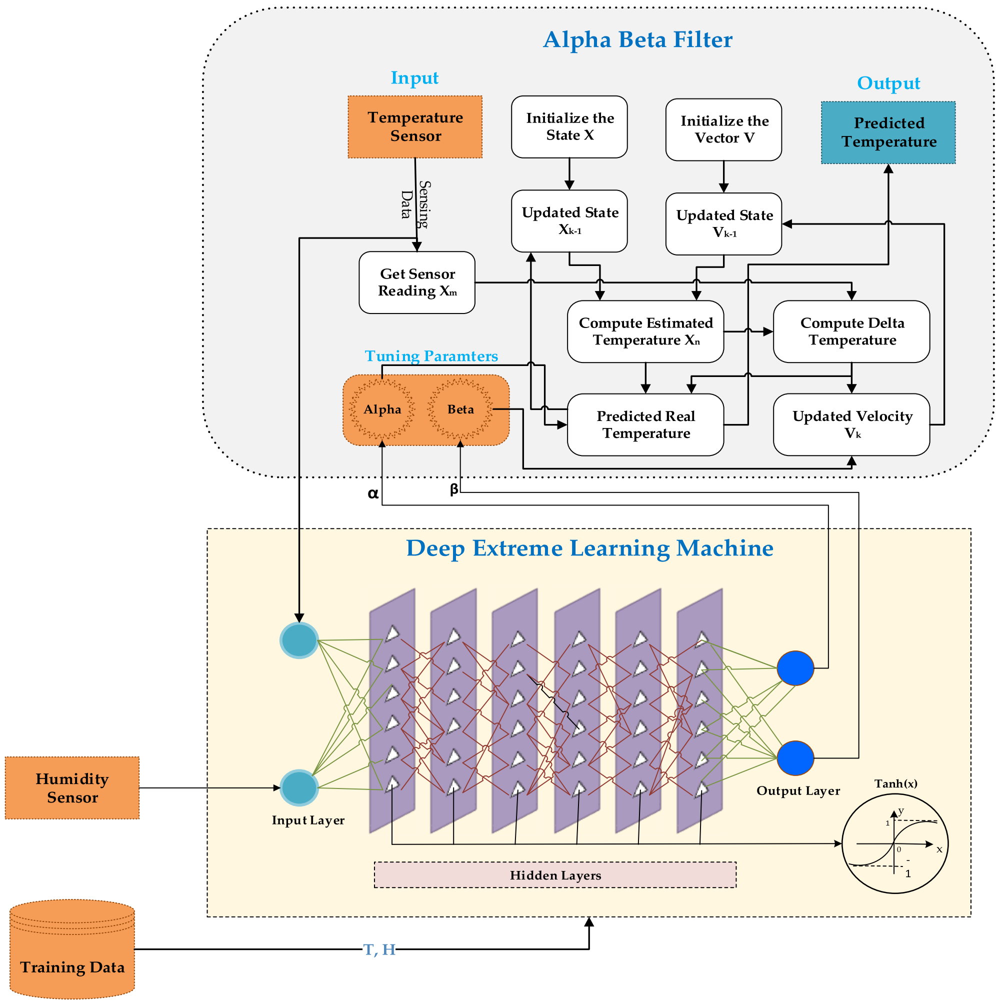


FIGURE 3. Detail proposed diagram of learning to an alpha-beta filter model.

$$B=\begin{bmatrix} b_{11} & b_{12} & \dots & b_{1z} \\ b_{21} & b_{22} & \dots & b_{2z} \\ b_{31} & b_{32} & \dots & b_{3z} \\ \vdots & \vdots & & \vdots \\ b_{r1} & b_{r2} & \dots & b_{rz} \end{bmatrix} \tag{2}$$

$$w=\begin{bmatrix} w_{11} & w_{12} & \dots & w_{1p} \\ w_{21} & w_{22} & \dots & w_{2p} \\ w_{31} & w_{32} & \dots & w_{3p} \\ \vdots & \vdots & & \vdots \\ w_{i1} & w_{i2} & \dots & w_{ip} \end{bmatrix} \tag{3}$$

$$\gamma=\begin{bmatrix} \gamma_{11} & \gamma_{12} & \dots & \gamma_{1r} \\ \gamma_{21} & \gamma_{22} & \dots & \gamma_{2r} \\ \gamma_{31} & \gamma_{32} & \dots & \gamma_{3r} \\ \vdots & \vdots & & \vdots \\ \gamma_{p1} & \gamma_{p2} & \dots & \gamma_{pr} \end{bmatrix} \tag{4}$$

Then, in the hidden layers, the extreme learning machine's biases are selected arbitrarily, as shown in (5). Next, $g(x)$ function is carried by the ELM that is the activation

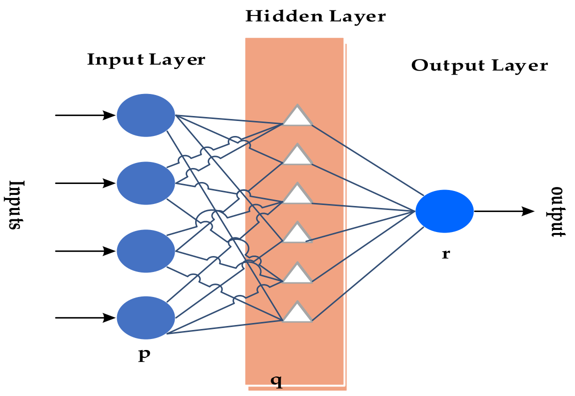


FIGURE 4. Structure diagram of ELM.

function of the whole network for ELM as shown in Figure 4, where the resultant matrix is depicted in (6), and the resultant matrix T of the column vector is depicted in (7) respectively.

$$B=[b_1,b_2,b_3,\dots,b_p]^T \tag{5}$$

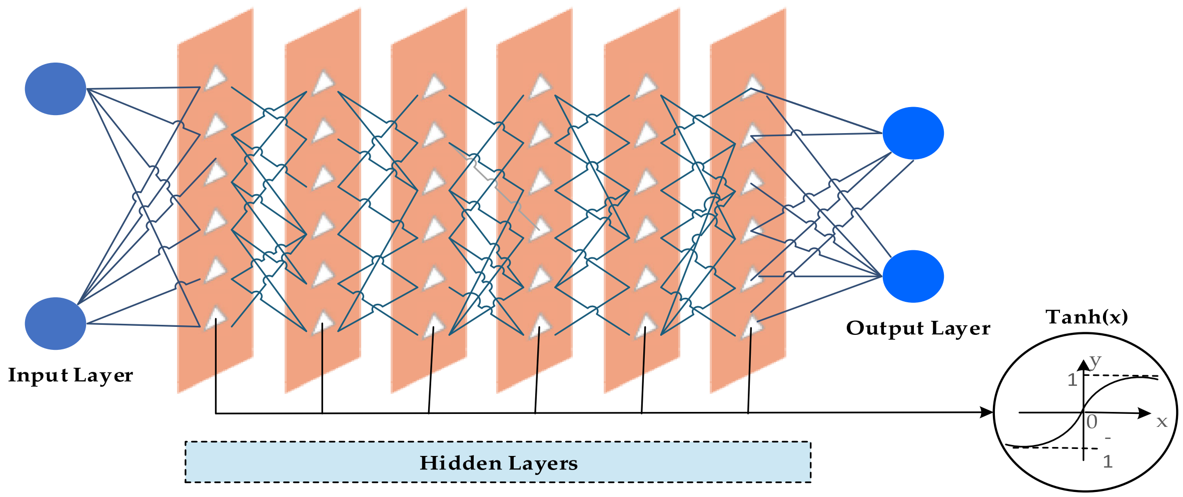


FIGURE 5. Deep extreme learning machine.

$$V = \left[v_1, v_2, v_3, \cdots, v_Z\right]_{r\times Z} \tag{6}$$

$$v_l = \begin{bmatrix} v_{1j} \\ v_{2j} \\ t_{3j} \\ \vdots \\ t_{rj} \end{bmatrix} = \begin{bmatrix} \sum_{l=1}^{q} \gamma_{k1} g(w_k a_l + b_k) \\ \sum_{l=1}^{q} \gamma_{k2} g(w_k a_l + b_k) \\ \sum_{l=1}^{q} \gamma_{k3} g(w_k a_l + b_k) \\ \vdots \\ \sum_{l=1}^{q} \gamma_{kr} g(w_k a_l + b_k) \end{bmatrix}_{(l=1,2,3,\ldots,Q)} \tag{7}$$

If we compute (6) and (7), the resulted equation is obtained in (8). The hidden layer output is denoted by H, V' is the transposition of V, and the weight matrix values were calculated using the least square method, which is denoted by γ, as shown in (9) [47],[48].

$$\mathrm{H}\gamma = V' \tag{8}$$
$$\gamma = \mathrm{H}^{+}V' \tag{9}$$

Accordingly, to make the network more stabilized and more generalized, the term regularization γ has been used, as in [49].

Nowadays, deep learning is the most popular, emerging, and widely used topic for researchers. DL network with input/output layers and having at least four layers meet the DLN criteria. The deep learning network neurons are trained in each layer, having a different type of constraints and parameters using the previous layers' output. DL networks also enable researchers to handle a widespread large variety of data sets. Because of the efficiency of deep learning, it grasped the attention to solve real-world problems that are very complex too. In the proposed methodology, we have combined DL, and ELM features to encapsulate the advantages of both and developed a new method called DELM [20].

The structure model of DELM is shown in Figure 5; the model of DELM has two neurons in the input layer, six different hidden layers, and every single hidden layer comprised of ten neurons and two neurons in a single output layer. Due to the lack of any technique to identify hidden layers neurons, how many neurons should be used in the hidden layers, and how many nodes should be selected, we used the technique called trial-and-error. To calculate the output neuron in the 2nd hidden layer (10) has been used.

$$H_1 = V\gamma^{+} \tag{10}$$

where the general inverse of matrix γ is denoted by γ^{+}. Whereas the 2nd hidden layer values can be normally computed in (11), and the activation function inverse can also be used.

$$H_1 = V\gamma^{+} \tag{11}$$
$$g\left(W_1 H + B_1\right) = H_1 \tag{12}$$

The parameters used in the (12) are defined as; the weighted matrix is denoted by W_1 of the first two hidden layers of the total network, H represents the biasness of the first neuron's hidden-layer. The predictable output of the first and the second hidden-layer are denoted by B_1 and H_1 respectively.

$$W_{HE} = g^{-1}(H_1)H_E^{+}, \tag{13}$$

The inverse of H_E is represented by H_E^{+} and g(x) is the activated function used to signifies (3), and the desired output of the second hidden layer is updated to specify any appropriate activation function $g(x)$, as shown in (14).

$$H_2 = g\left(W_{HE} H_E\right) \tag{14}$$

To update the γ weight Matrix among hidden-layer ii and hidden layer iii has been performed using (15). Whereas the inverse of H_2 is indicated by H_2^{+} and the desired value of the hidden layer three is denoted in (16).

$$\gamma_{new} = H_2^{+}V \tag{15}$$
$$H_3 = V\gamma_{new}^{+} \tag{16}$$

The inverse of the weight matrix γ_{new} is represented by $V\gamma_{new}^{+}$. Subsequently, DELM defines the matrix $W_{HE1} = [B_2, W_2]$, and output results of the third layer are accomplished through (8 and 9).

$$H_3 = g^{-1}(H_2W_2 + B_2) = g(W_{HE1}H_{E1}) \tag{17}$$
$$W_{HE1} = \gamma^{-1}(H_3)H_{E1}^{+}) \tag{18}$$

If we discuss the rest of the equations, the inverse of the activation function $g(x)$ is denoted by $g^{-1}(x)$. The hidden layer two and desired result is denoted by H_2, W_2 represent the weight between the hidden layer two and three, where the biasness of the hidden layer three neurons is signified by B_2. Whereas the inverse of H_{E1} is represented by H_{E1}^{+}. The logistic sigmoid function has been adopted as represented in (19). Now we compute the output of the 3rd hidden layer, as shown in (20).

$$g(x) = \frac{1}{1+e^{-x}} \tag{19}$$
$$H_3 = g(W_{HE1}H_{E1}) \tag{20}$$

The final weighted resultant matrix of the hidden-layer three and the final layer's output is calculated, so the expected output of hidden layer three is also calculated; both are shown in (21,22).

$$\gamma_{new} = H_4^T(\frac{1}{\lambda} + H_4^TH_4)^{-1} V \tag{21}$$
$$H_4 = V\gamma_{new}^{+} \tag{22}$$

The inverse of the weight matrix γ_{new} is represented by $V\gamma_{new}^{+}$. Subsequently, DELM defines the matrix $W_{HE2} = [B_3, W_3]$, and the fourth layer's output result is accomplished by the above equations (10 and 19).

$$H_4 = g^{-1}(H_3W_3 + B_3) = g(W_{HE1}H_{E1}) \tag{23}$$
$$W_{HE2} = \gamma^{-1}((H_4)H_{E2}^{+} \tag{24}$$

The inverse of activation function $g(x)$ is denoted by $g^{-1}(x)$ as shown in (9), H_3 represents the output required for the 3rd hidden-layer network, W_3 represent the weight between the 3rd layer and the 4th hidden layers, where the biasness of the 3rd hidden layer neurons is signified by B_3. H_{E1}^{+} denotes the opposite of H_{E1}.The logistic sigmoid function has been adopted as described. Now we compute the fourth hidden layer, as shown in (25).

$$H_4 = g(W_{HE2}H_{E2}) \tag{25}$$

Finally, in (26), the matrix weight output between the 4th and the output layers is calculated. The predictable output of the fifth hidden layer is denoted in (27), and the desired result of the DELM network is shown below in (28).

$$\gamma_{new} = H_5^T(\frac{1}{\lambda} + H_5^TH_5)^{-1} V \tag{26}$$
$$H_5 = V\gamma_{new}^{+} \tag{27}$$
$$f(x) = H_5\beta_{new} \tag{28}$$

In conclusion, we have briefly discussed the complete mathematical calculation of the DELM network for four hidden layers. To get the parameters and store individual hidden layers parameters, (16-20) has been recalculated and ultimately recorded the final output of the DELM network. For further computation and calculation of the network, we have applied the cycle theory. If the proposed computation gives higher performance accuracy in the hidden layers network, then a similar calculation practice can be recycled and executed correspondingly to gain maximum performance. The trial and error method [50], [51] in the proposed work is used to generate an optimal neural network model.

B. ALPHA-BETA (α-β) FILTER

α-β filter is a simplified form of observer typically used for estimation, control applications, and smoothing. The functionality of the α-β filter is the same as the linear observers and the Kalman filter algorithm. The key benefit of the α-β filter is its Simplicity because it does not need a comprehensive model for learning. It is derived from the Kalman filter algorithm [1]. α-β filter needs very low space, and computation time as associated with the Kalman filter algorithm, the performance of α-β filter is also improved as related to the linear filter.

If we derive the α-β filter, then we have to presume that a system is adequately estimated by a model with two interval positions. In the first step, we can do the initialization, as shown in Equations (29, 30) [52]–[54].

$$x_{k-1} = c_1 \tag{29}$$
$$v_{k-1} = c_2 \tag{30}$$

Equation (31) is used to update the position, and for reading sensor data, (32) is used.

$$x_k = x_{k-1} + v_{k-1}.\Delta t \tag{31}$$
$$x_m = Sensor() \tag{32}$$

For computing, the difference (33) is used, and for computing predicting position (34) has been used.

$$\Delta x_k = x_m - x_k \tag{33}$$
$$x_k = x_k + \alpha \cdot \Delta x_k \tag{34}$$

For computing predicted velocity (35) is used, we also have used (36-37) for updating the position & velocity for the next iteration.

$$v_k = v_{k-1} + \beta.\Delta x_k/\Delta t \tag{35}$$
$$x_{k-1} = x_k \tag{36}$$
$$v_{k-1} = v_k \tag{37}$$

The configuration model of the α-β filter is shown in Figure 6.

C. PERFORMANCE EVALUATION METHOD

There are a different number of parameters available to compute the performance of the algorithm, but three of them

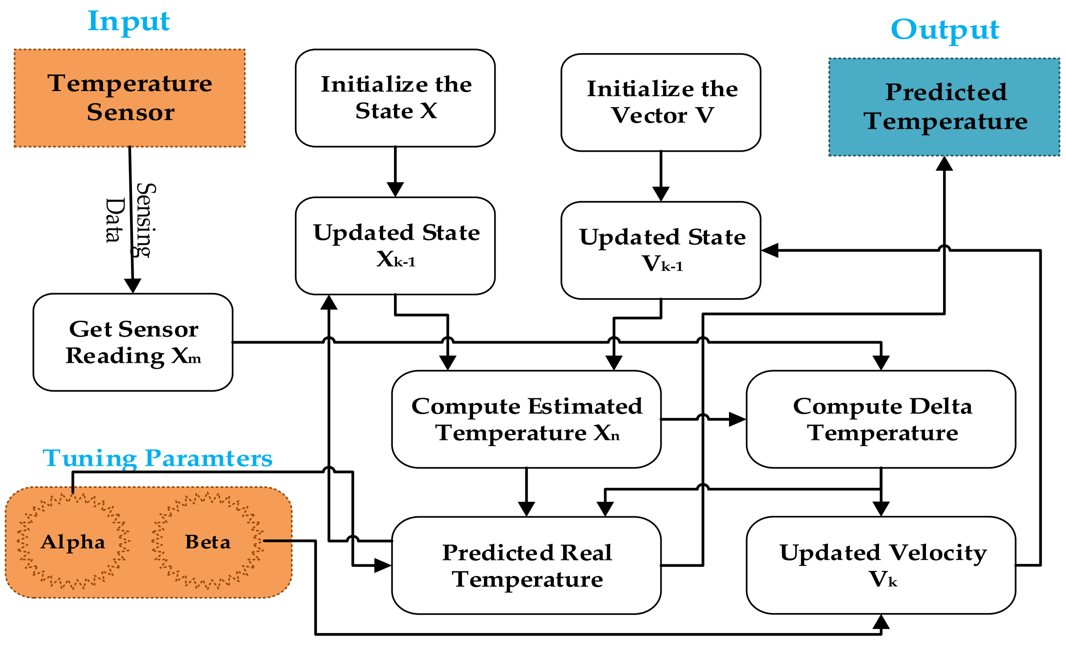


FIGURE 6. Configuration diagram of alpha-beta filter.

we have used. In our proposed work, to calculate the performance of the algorithm, the three performance-evaluation methods are considered: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Square Error (MSE) [1], [11]. These three methods are used for the measurement of the proposed approach. The discussed performance measurement is calculated using the Equations (38-40), respectively.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum\nolimits_{k=0}^{n}(A-P_i)^2} \tag{38}$$

$$\mathrm{MAE} = \frac{1}{N}\sum\nolimits_{i=1}^{n}|A_i-P_i| \tag{39}$$

$$\mathrm{MSE} = \frac{1}{N}\sum\nolimits_{k=0}^{n}(A-P_i)^2 \tag{40}$$

In the above three equations, A signifies the actual values, N represents the total number of observations, and P denotes the predicted values.

IV. EXPERIMENTAL SETUP, IMPLEMENTATION RESULTS, AND PERFORMANCE EVALUATION

A. EXPERIMENTAL SETUP

For experimental analysis and performance evaluation of the alpha-beta filter, we have used a collected actual dataset of Seoul containing temperature and humidity data and simulated noisy sensor readings for the one-year duration in Seoul. Figure 8 (a, b) represents the actual temperature data and the humidity data collected from January-December, 2010, over every day's hourly interval [1]. The total number of days in a year is 365, temperature values for 12 months are 12, and for humidity for 12 months is 12, which both are 24. There were 8760 total data instances because of $365 \times 24 = 8760$ samples. The correlation between the actual temperature values and humidity values has been calculated using the Pearson correlation coefficient (CC) as denoted in (41).

$$\mathrm{CC\ (T,H)} = \rho = \frac{\sum(t_i-\bar{t})(h_i-\bar{h})}{\sqrt{\sum(t_i-\bar{t})^2\sum(h_i-\bar{h})^2}} \tag{41}$$

where the correlation coefficient (CC) between temperature and humidity is denoted by CC (T, H) also denoted by ρ, and t_i represent the temperature values and h_i denotes the values of humidity in the i$^{\text{th}}$ hour. The mean temperature values are represented as $\bar{t}$ and mean humidity values are represented by $\bar{h}$.

A significant positive correlation between the actual temperature and the humidity level occurs and $r(8758) = 0.22$, where $p < 0.0001$, but weak correlation. For the generation of varying conditions and scenarios dynamically, the error has been added into the sensor readings of temperature based upon humidity level, and here a uniform distribution has been used. The total quantity of error was arbitrarily created, but the stabilized current humidity level was proportional, as shown in (42).

$$|Err| = \frac{h_{cur}-h_{min}}{h_{max}-h_{min}} \tag{42}$$

where the absolute error is represented by $|Err|$ in the temperature sensor readings, the current humidity level is represented by h_{cur}, the maximum humidity level is denoted by h_{max}, and minimum humidity level is denoted by h_{min}. Equation (43) has been used to calculate the simulated sensor

TABLE 2. Different configurations of training and testing with the help of 4-folds cross-validation method prediction & results shows with the help of RMSE.

DELM Configuration		Model-1		Model-2		Model-3		Model-4		MA	Experiments Average
HL	AF	Training	Testing	Training	Testing	Training	Testing	Training	Testing		
5	Sigmoid	4.43	5.09	4.99	3.20	4.40	4.98	4.41	4.90	4.55	
5	Tanh	3.44	3.23	4.40	3.33	3.78	3.07	4.687	3.864	3.72	4.204
5	Linear	4.09	5.11	4.65	3.45	4.69	3.88	4.98	3.9	4.34375	
10	Sigmoid	4.22	4.32	3.21	3.11	4.21	3.55	3.31	4.32	**3.78**	
10	**Tanh**	**3.67**	**3.31**	**1.4**	**1.3**	**2.43**	**3.34**	**.987**	**.864**	**2.163**	**3.296**
10	Linear	3.09	4.14	3.43	3.45	4.69	3.88	4.98	3.9	**3.945**	
15	Sigmoid	4.22	4.32	3.21	3.11	4.21	3.55	3.31	4.32	3.781	
15	Tanh	2.67	2.31	2.3	2.31	2.43	2.34	2.987	2.84	2.523	3.41
15	Linear	3.09	4.14	3.43	3.45	4.69	3.88	4.98	3.9	3.945	
20	Sigmoid	4.30	4.99	4.87	3.83	4.23	4.80	4.56	4.93	4.564	
20	Tanh	.67	2.31	2.3	2.31	2.43	2.34	2.987	2.84	2.2734	3.73
20	Linear	4.09	5.11	4.65	3.45	4.69	3.88	4.98	3.9	4.344	

HL: Hidden Layers, **AF:** Activation Function, **MA:** Models Average, **DELM**: Deep Extreme Learning Machine, and **Tanh**: hyperbolic tangent function

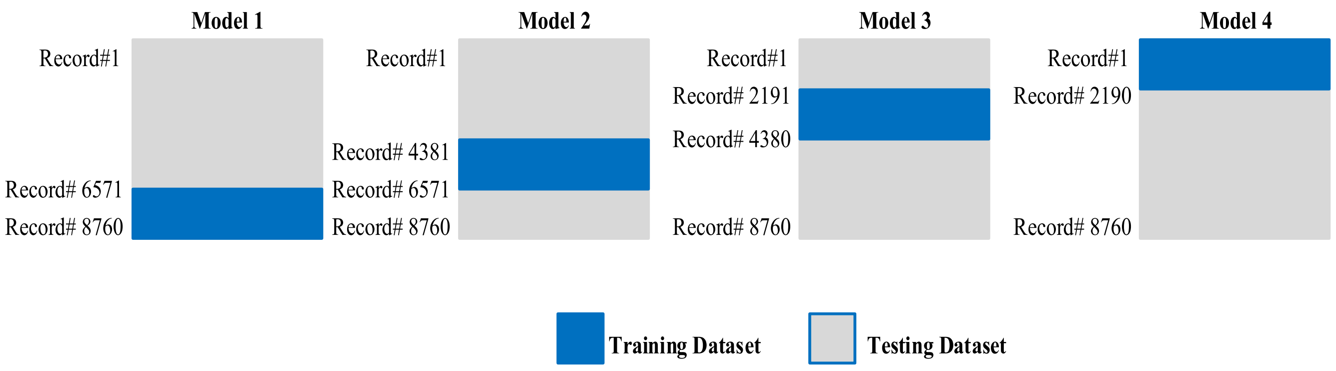


FIGURE 7. 4-folds cross-validation model for training and testing datasets.

readings with noise.

$$|Err| = \frac{h_{cur} - h_{min}}{h_{max} - h_{min}} \times \mathfrak{R}(-1, 1) \times \mathrm{S} + \mathrm{T}_{\mathrm{org}} \tag{43}$$

where the simulated noise sensor reading is represented by T_{sen} to produce arbitrary values in the range of -1 and $+1$, $\mathfrak{R}$ is used with a uniform distribution, the scaling factor of the error is represented by S, and T_{org} denotes the original temperature.

An appropriate number of hidden layers and the quantity of neurons in the hidden layers in a network is a trial-and-error type of practice. In the proposed work, we have tried a different number of hidden layers and a different number of neurons in hidden layers to select the appropriate number of hidden layers and the number of neurons in hidden layers. Similarly, we have also applied different activation functions to select the proposed model's appropriate activation function.

In the proposed work, we have considered 10 hidden layers and a hyperbolic tangent function as an activation function because the RMSE value for this configuration is low related to the other configurations, as shown in Table 2. The network's initialization is carried out with random weights because there is no proper mechanism to initialize the proper weight to the network [2]. To avoid overfitting in the training processes, different configurations and 4-folds cross-validation methods have been used. The dataset has been divided into four subsets of equal size. In Figure 7, the training and testing of the dataset used for each model use 4-folds cross-validation process. As we have used the 4 folds cross-validation model in the proposed work, in each fold, 75% of data are used for training and 25% for testing. The total number of epoch used to test the algorithm is 100.

In the proposed DELM, we have used the random weight technique [14].

To select the best configuration, we have used the configuration of DELM and the predicted accuracy of each configuration and recorded in terms of RMSE for training and testing, as given in Table 2.

B. IMPLEMENTATION RESULTS

In the proposed work, we have two main modules learning and prediction; the implementation of both modules has been done on Intel(R) Core (TM) i3-4010U with RAM 4.00 GB and CPU @ 1.70GHz and MATLAB version R2018a. The dataset comprised hourly temperature, humidity, and simulated noisy sensor readings data for a one-year duration. The data are comprised of four different input parameters; one

TABLE 3. Summarization of gathered data and noisy simulated data.

Measure	Temperature (°C)	TSR	Humidity (g/ m³)	AETSR
Min	-15.30	-21.30	12	0
Max	33.40	38.58	97	10
Avg	12.14	12.16	62.90	5.99
SD	11.39	12.58	19.89	2.34

AETSR: Absolute Error in Temperature Sensor Reading, **TSR**: Temperature Sensor Reading, Avg: Average, SD: Standard Deviation

is original temperature values, the second is humidity level parameters, third is the noisy sensor reading data, and the last is the number of sensor readings, as illustrated in Figure 8. The root mean square error for the readings sensor is 5.21, which is very high.

The given results show that with the Tanh activation function, the results of DELM are rarely affected by increasing and decreasing the number of neurons in the hidden layer. However, the remarkable difference in the forecasting accuracy can be noticed with an individual model in the 4-folds cross-validation process. But an interesting fact is that in the case of the Model-2, higher forecasting accuracy is attained with the testing of the Model-2 as compared with the other training datasets. The Tanh activation function is used in DELM, and remarkable enhancement in the prediction accuracy can be noticed in the given results as compared to the linear AF as shown in Table 2. The best-case accuracy was achieved for the alpha-beta filter with DELM having ten neurons in the hidden layer and Tanh as an activation function.

Further, we have calculated the statistical measures to indicate the variation between original temperature values and noisy sensor reading values. The summarization of the gathered data and noisy simulated data is given in Table 3.

The alpha-beta filter has been applied because the variation between sensor temperature values and the original values is very high, as shown in Figure 9. The MAE and RMSE, and MAPE values are 3.9515, 5.2158, and 27.2046, respectively. For the alpha-beta filter, we have included the learning unit to increase the efficiency of the conventional α-β filter's performance.

In the proposed work, we have used the DELM for tuning the alpha-beta values based on the previous data, as shown in Figure 10. The sample values of alpha-beta are given in Table 3. To increase the DELM performance, we have normalized the input and output data and feed it to the DELM. The output of the DELM is then deformalized to get the original alpha-beta values. We have used (44) for normalization and (45) for de-normalization.

$$Nor(i) = \frac{\text{curr (i)} - \min(i)}{\max(i) - \min(i)} \tag{44}$$

$$D\_Nor(i) = Nor\ (i) \times (\max(i) - \min(i)) + \min(i) \tag{45}$$

where $Nor(i)$ represents the desire normalized data, curr(i) indicates the current value that is to be normalized, min(i) represents the minimum values and max (i) represents the maximum values in the data, respectively.

The sensor reading values, the conventional alpha-beta predicted temperature values, and the original temperature values are shown in Figure 11. Similarly, sensor reading values, the learning to alpha-beta filter predicted temperature values, and the original temperature values are presented in Figure 12. The variation between predicted learning to alpha-beta filter temperature values and the original temperature values is less than the conventional alpha filter predicted values. The original temperature values indicate the performance of the learning to the α-β filter is outperformed as compared to the conventional alpha-beta filter values. We have also taken sample values and shown them as subfigures in Figures (11 and 12) to exhibit the results better.

The MAE, RMSE, MSE have been calculated for the conventional α-β filter and proposed learning to alpha-beta filter as listed in Table 4 and Figure 13. The comparative enhancement in the prediction accuracy of the proposed learning-to-prediction model was 7.72% −16.47% in terms of RMSE and MSE metrics. The values of statistical measures represent that the performance of our proposed learning to alpha-beta filter is too much better as compared with the conventional alpha-beta filter that validates that the proposed technique outperforms the conventional α-β filter.

TABLE 4. Statistical measures for learning and conventional α-β filter.

Algorithms	Proposed Alpha Beta Filter	Conventional Alpha Beta Filter
MAE	2.3586	2.556
RMSE	2.9614	3.240
MSE	8.7699	10.5

MAE: Mean absolute Error, **RMSE:** Root Mean Square Error, **MSE:** Mean Square Error

C. COMPARISON OF PROPOSED ALPHA BETA FILTER WITH EXISTING KALMAN FILTER ALGORITHM

We have already discussed in our proposed methodology that the α-β filter is resulting from the Kalman filter algorithm and the performance of the α-β filter is outperform

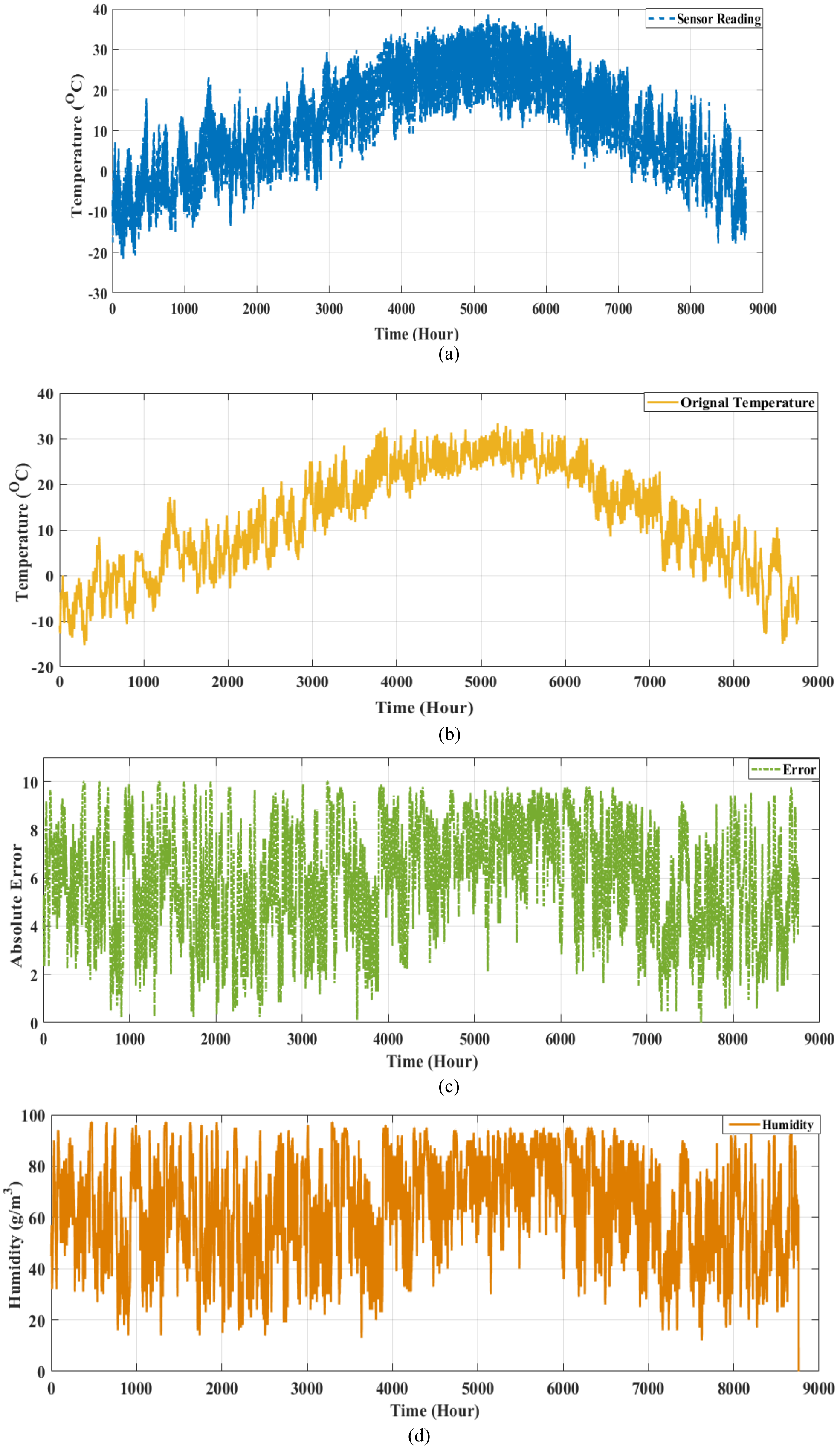


FIGURE 8. Input parameters (a) temperature, (b) noisy sensor reading, (c) absolute error, and (d) humidity level.

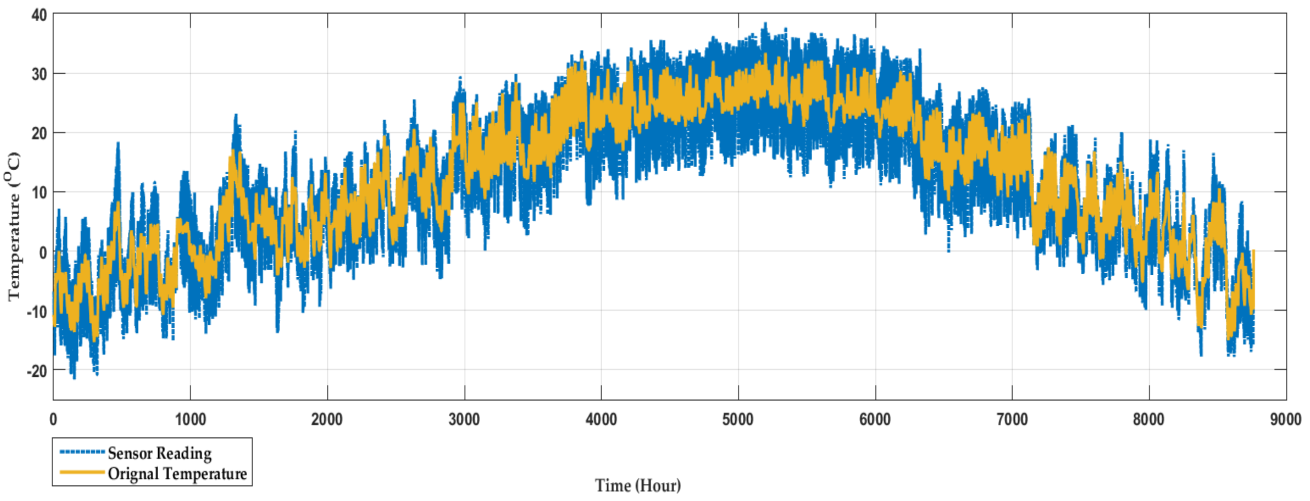


FIGURE 9. Original temperature and simulated noisy sensor reading data.

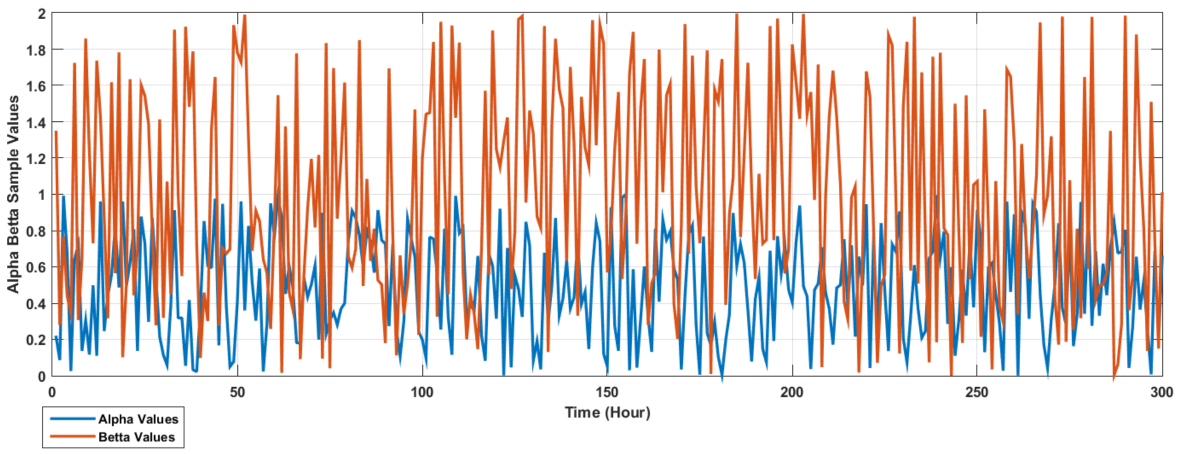


FIGURE 10. Sample values of alpha-beta for deep extreme learning machine training.

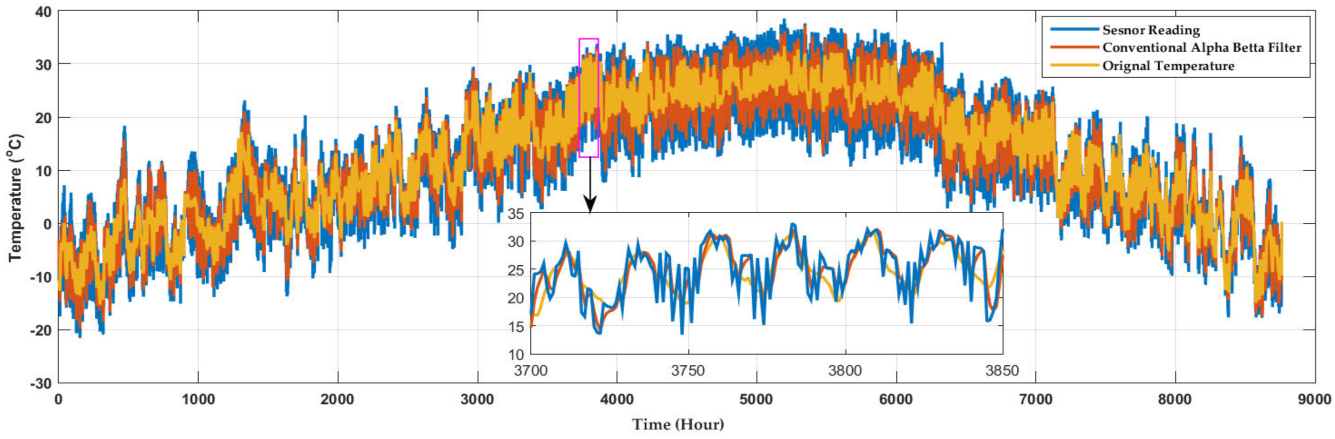


FIGURE 11. Original temperature, sensor reading, and predicted alpha-beta filter temperature values.

as compared to the Kalman filter algorithm because in the Kalman filter algorithm, every time we need to calculate the Kalman gain, but alpha-beta filter does not require Kalman gain. Simplicity is the key feature of the α-β filter. If we

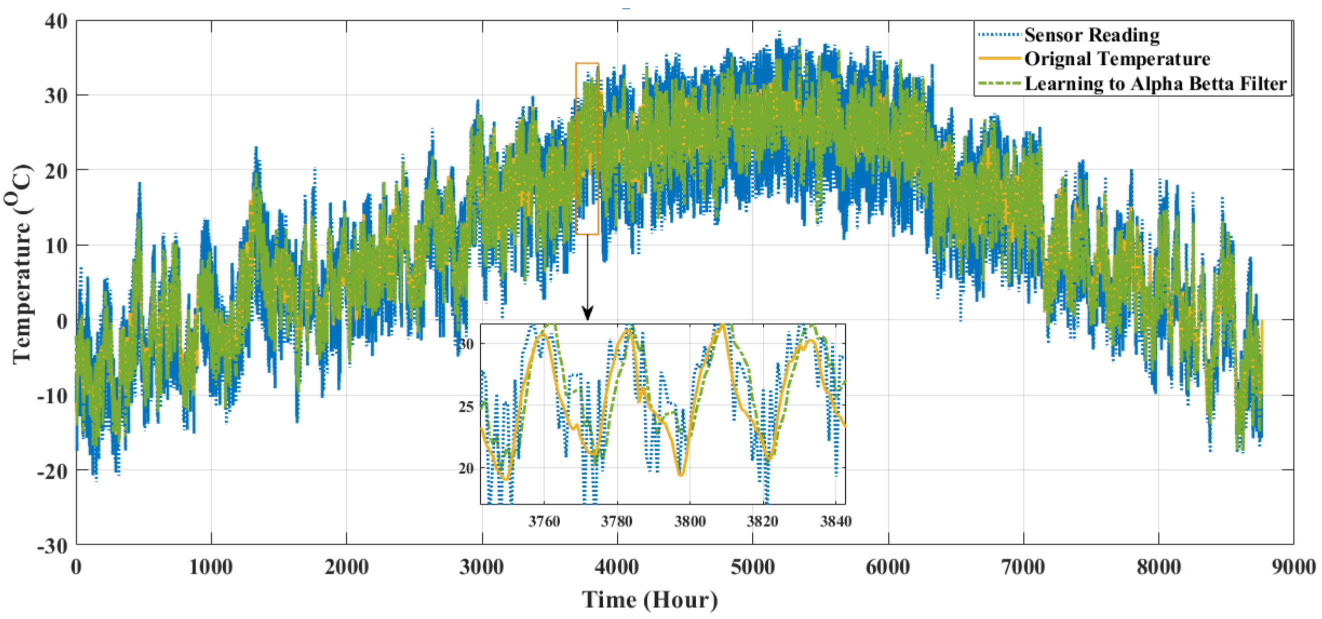


FIGURE 12. Original temperature, sensor reading, and predicted learning to alpha-beta filter temperature values.

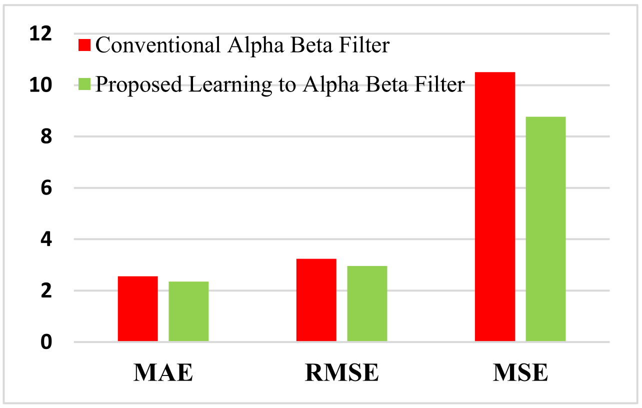


FIGURE 13. omparison of proposed and conventional alpha beta filter.

TABLE 5. Comparison of the proposed method with existing method in terms of RMSE and MSE.

Performance Metrics	Kalman Filter With the Learning Module	Proposed Learning to Alpha-Beta Filter
RMSE	4.41%	7.72%
MSE	11.19%	16.47%

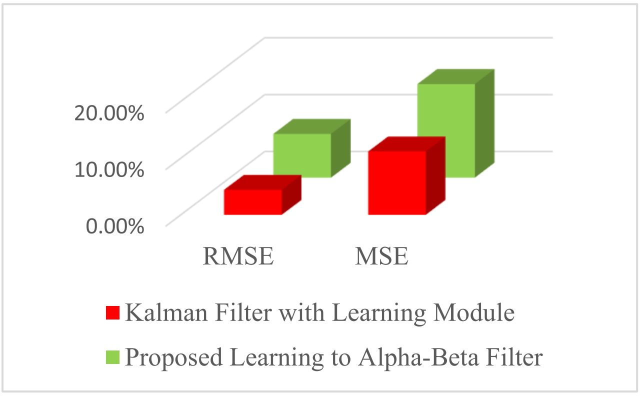


FIGURE 14. omparison of the proposed learning to alpha-beta filter with existing kalman filter in terms of performance increase.

compare the results of both the Kalman filter algorithm and alpha-beta filter with learning modules added to it and show the performance in terms of RMSE and MSE. The Kalman filter relative performance improvement was 4.41-11.19% with the learning module added. If we can see the relative performance improvement of the α-β filter with the learning unit, this performance can jump to 7.72-16.47%, as shown in detail in Table 5 and Figure 14.

V. DISCUSSION

The accuracy of the algorithms depends on the data provided to them, and for reliable data, the filters play an essential role. The tuning of the parameters is necessary for reliable data.

For experimental analysis of the algorithm, the α-β filter has been used to predict the parameters as a prediction algorithm. The conventional alpha-beta filter flops to predict real parameters in dynamic situations. Therefore, the proposed improved learning to prediction algorithm improves the performance of the alpha-beta filter algorithm.

In the conventional alpha-beta filter, the alpha filter and beta filter values are fixed. The fixation of values is the main weakness of this algorithm because different alpha-beta filter provides different results on different alpha-beta filter values. Thus, main objective of this work is to tune the alpha-beta values based on some previous experience for every input. In our proposed model, we have used the learning module comprised of DELM for tuning the alpha-beta values to increase the alpha-beta filter's performance.

The basic objective was to use historical data to make the alpha-beta values dynamic in the proposed work. We used the DELM algorithm, a machine learning approach, to tackle this problem in this work. First, we trained the DELM on historical data and then used the current inputs from the environment to the DELM algorithm and alpha-beta filter. Based on the previous training data, the DELM set the alpha-beta values for each input of the environment and provide the output.

In the proposed work to set the appropriate number of hidden layers, the number of neurons in hidden layers, and the appropriate type of activation function, we have used the trial-and-error method. We applied a different number of hidden layers with different neurons in the hidden layers and different types of activation functions and recorded the results. Then we select the best combination of the hidden number of layers, hidden number of neurons in hidden layers, and activation function.

Different authors have also used many other techniques, such as fuzzy inference, to tune the alpha-beta filter values in the alpha-beta filter. However, these techniques are not effective because the alpha-beta values are provided through fuzzy logic, but these values are randomly assigned to alpha-beta parameters in the alpha-beta filter.

The proposed approach has been evaluated using different performance measures, and the results are satisfactory compared to the conventional alpha-beta filter algorithm. We can improve the results more improve as the number of instances in the historical dataset increase.

In the proposed work, we have considered less number of parameters, and the data size is also small, and these are the main limitation of the proposed work. In the future, we will consider more parameters such as illumination, air quality.

Another fact is that a lot of machine learning algorithm exists and there is still a need to test the alpha-beta filter with other machine learning algorithm and compare the result of the different algorithm. In the future, more methodology can be explored and incorporated into the proposed work.

VI. CONCLUSION

This paper focuses on the learning to prediction method to boost the algorithms' prediction performance in dynamic circumstances. This paper has proposed a new method based upon the alpha-beta filter and DELM named learning to alpha beta filter. The proposed method consists of two main components, namely the prediction unit and the learning unit. In our prediction component, we have used the α-β filter, and in the learning unit, the DELM has been used. The conventional alpha-beta filter's main problem is that alpha-beta values are typically selected using the trial-and-error approach. Once the alpha-beta values are chosen for a specific problem, they remain fixed for the entire data. It is observed that different alpha-beta values for the same problem give different results. Hence, in the proposed method, we have addressed this problem and added the learning module to the conventional alpha-beta filter to improve its performance. The relative improvement in the prediction accuracy of the proposed learning to prediction model was 7.72% and 16.47% in terms of RMSE and MSE metrics. We have applied the proposed learning to the alpha-beta filter on the data, and the results show that the performance of the proposed learning to the alpha-beta filter is too much better than the conventional alpha-beta filter.

REFERENCES

[1] I. Ullah, M. Fayaz, and D. Kim, "Improving accuracy of the Kalman filter algorithm in dynamic conditions using ANN-based learning module," *Symmetry*, vol. 11, no. 1, p. 94, Jan. 2019. [Online]. Available: https://www.mdpi.com/2073-8994/11/1/94

[2] A. Shah, H. Nasir, M. Fayaz, A. Lajis, and A. Shah, "A review on energy consumption optimization techniques in IoT based smart building environments," *Information*, vol. 10, no. 3, p. 108, Mar. 2019, doi: 10.3390/info10030108.

[3] F. Wahid, R. Ghazali, M. Fayaz, and A. S. Shah, "Statistical features based approach (SFBA) for hourly energy consumption prediction using neural network," *Int. J. Inf. Technol. Comput. Sci.*, vol. 9, no. 5, pp. 23–30, May 2017, doi: 10.5815/ijitcs.2017.05.04.

[4] F. Wahid, R. Ghazali, M. Fayaz, and A. S. Shah, "A simple and easy approach for home appliances energy consumption prediction in residential buildings using machine learning techniques," *J. Appl. Environ. Biol. Sci.*, vol. 7, no. 3, pp. 108–119, 2017.

[5] S. R. Gunn, "Support vector machines for classification and regression," *ISIS Tech. Rep.*, vol. 14, no. 1, pp. 5–16, 1998.

[6] Z. Zhang, K. Zhang, and A. Khelifi, *Multivariate Time Series Analysis in Climate and Environmental Research*. Springer, 2018.

[7] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.

[8] M. Fayaz, H. Shah, A. Aseere, W. Mashwani, and A. Shah, "A framework for prediction of household energy consumption using feed forward back propagation neural network," *Technologies*, vol. 7, no. 2, p. 30, Apr. 2019.

[9] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1177–1193, Aug. 2012. [Online]. Available: https://ieeexplore.ieee.org/document/6215056/

[10] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Jan. 1992.

[11] M. Fayaz and D. Kim, "A prediction methodology of energy consumption based on deep extreme learning machine and comparative analysis in residential buildings," *Electronics*, vol. 7, no. 10, p. 222, Sep. 2018.

[12] S. Ding, N. Zhang, X. Xu, L. Guo, and J. Zhang, "Deep extreme learning machine and its application in EEG classification," *Math. Problems Eng.*, vol. 2015, p. 11, 2015, Art no. 129021, doi: 10.1155/2015/129021.

[13] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2004, pp. 985–990.

[14] W. Cao, X. Wang, Z. Ming, and J. Gao, "A review on neural networks with random weights," *Neurocomputing*, vol. 275, pp. 278–287, Jan. 2018.

[15] W. Cao, J. Gao, Z. Ming, and S. Cai, "Some tricks in parameter selection for extreme learning machine," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 261, Oct. 2017, Art. no. 012002.

[16] W. Cao, J. Gao, Z. Ming, S. Cai, and H. Zheng, "Impact of probability distribution selection on RVFL performance," in *Proc. Int. Conf. Smart Comput. Commun.*, Springer, 2017, pp. 114–124.

[17] W. Cao, M. J. A. Patwary, P. Yang, X. Wang, and Z. Ming, "An initial study on the relationship between meta features of dataset and the initialization of NNRW," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[18] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: A survey," *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 2, pp. 107–122, Jun. 2011.

[19] W. Cao, Z. Ming, Z. Xu, J. Zhang, and Q. Wang, "Online sequential extreme learning machine with dynamic forgetting factor," *IEEE Access*, vol. 7, pp. 179746–179757, 2019.

[20] A. S. Shah, H. Nasir, M. Fayaz, A. Lajis, I. Ullah, and A. Shah, "Dynamic user preference parameters selection and energy consumption optimization for smart homes using deep extreme learning machine and bat algorithm," *IEEE Access*, vol. 8, pp. 204744–204762, 2020, doi: 10.1109/ACCESS.2020.3037081.

[21] C. W. Kang and C. G. Park, "Attitude estimation with accelerometers and gyros using fuzzy tuned Kalman filter," in *Proc. Eur. Control Conf. (ECC)*, Aug. 2009, pp. 3713–3718.

[22] M. N. Ibarra-Bonilla, P. J. Escamilla-Ambrosio, and J. M. Ramirez-Cortes, "Attitude estimation using a neuro-fuzzy tuning based adaptive Kalman filter," *J. Intell. Fuzzy Syst.*, vol. 29, no. 2, pp. 479–488, Oct. 2015.

[23] H. Rong, C. Peng, Y. Chen, L. Zou, Y. Zhu, and J. Lv, "Adaptive-gain regulation of extended Kalman filter for use in inertial and magnetic units based on hidden Markov model," *IEEE Sensors J.*, vol. 18, no. 7, pp. 3016–3027, Apr. 2018.

[24] J. Havlík and O. Straka, "Performance evaluation of iterated extended Kalman filter with variable step-length," *J. Phys., Conf. Ser.*, vol. 659, Nov. 2015, Art. no. 012022.

[25] J. Huang, A. B. McBratney, B. Minasny, and J. Triantafilis, "Monitoring and modelling soil water dynamics using electromagnetic conductivity imaging and the ensemble Kalman filter," *Geoderma*, vol. 285, pp. 76–93, Jan. 2017.

[26] D. Połap, A. Winnicka, K. Serwata, K. Kęsik, and M. Woźniak, "An intelligent system for monitoring skin diseases," *Sensors*, vol. 18, no. 8, p. 2552, Aug. 2018.

[27] S. Zhao, Y. S. Shmaliy, P. Shi, and C. K. Ahn, "Fusion Kalman/UFIR filter for state estimation with uncertain parameters and noise statistics," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 3075–3083, Apr. 2017.

[28] M. Woźniak and D. Połap, "Adaptive neuro-heuristic hybrid model for fruit peel defects detection," *Neural Netw.*, vol. 98, pp. 16–33, Feb. 2018.

[29] M. Dahmani, A. Meche, M. Keche, and K. Abed-Meraim, "An improved fuzzy alpha-beta filter for tracking a highly maneuvering target," *Aerosp. Sci. Technol.*, vol. 58, pp. 298–305, Nov. 2016, doi: 10.1016/j.ast.2016.08.029.

[30] M. Abdelkrim, D. Mohammed, K. Mokhtar, and O. Abdelaziz, "A simplified $\alpha\beta$ based Gaussian sum filter," *AEU Int. J. Electron. Commun.*, vol. 67, no. 4, pp. 313–318, Apr. 2013, doi: 10.1016/j.aeue.2012.09.003.

[31] C.-M. Wu, C.-K. Chang, and T.-T. Chu, "A new EP-based $\alpha - \beta - \gamma - \delta$ filter for target tracking," *Math. Comput. Simul.*, vol. 81, no. 9, pp. 1785–1794, May 2011.

[32] F. Jamil and D. H. Kim, "Improving accuracy of the alpha–beta filter algorithm using an ANN-based learning mechanism in indoor navigation system," *Sensors*, vol. 19, no. 18, p. 3946, Sep. 2019.

[33] D. Yates, S. Gangopadhyay, B. Rajagopalan, and K. Strzepek, "A technique for generating regional climate scenarios using a nearest-neighbor algorithm," *Water Resour. Res.*, vol. 39, no. 7, pp. 1–15, Jul. 2003.

[34] M.-L. Zhang and Z.-H. Zhou, "A K-nearest neighbor based algorithm for multi-label classification," in *Proc. IEEE Int. Conf. Granular Comput.*, Jul. 2005, pp. 718–721.

[35] S. Suthaharan, "Machine learning models and algorithms for big data classification," *Integr. Ser. Inf. Syst*, vol. 36, pp. 1–12, Feb. 2016.

[36] I. Ullah, R. Ahmad, and D. Kim, "A prediction mechanism of energy consumption in residential buildings using hidden Markov model," *Energies*, vol. 11, no. 2, p. 358, Feb. 2018.

[37] M. Slocum, "Decision making using id3 algorithm," *Insight, River Academic J*, vol. 8, no. 2, pp. 1–12, 2012.

[38] H. Wang, H. Yi, J. Peng, G. Wang, Y. Liu, H. Jiang, and W. Liu, "Deterministic and probabilistic forecasting of photovoltaic power based on deep convolutional neural network," *Energy Convers. Manage.*, vol. 153, pp. 409–422, Dec. 2017.

[39] W.-L. Xu, N.-Y. Zhang, and K. Zeng, "A comparative study on sufficient conditions for Takagi-Sugeno fuzzy systems as universal approximators," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 6, pp. 773–780, Dec. 2000.

[40] I. Iancu, "A mamdani type fuzzy logic controller, fuzzy logic-controls," in *Concepts, Theories and Applications*. Rijeka, Croatia: InTech, 2012, pp. 325–350.

[41] G. Zahedi, S. Azizi, A. Bahadori, A. Elkamel, and S. R. W. Alwi, "Electricity demand estimation using an adaptive neuro-fuzzy network: A case study from the ontario province—Canada," *Energy*, vol. 49, pp. 323–328, Jan. 2013.

[42] Y.-M. Wang, J. Liu, T. M S Elhag, and L. M. Lopez, "Bridge risk assessment using a hybrid AHP/DEA methodology," in *Proc. Intell. Syst. Knowl. Eng. (ISKE)*, Oct. 2007, pp. 1561–1565.

[43] Y. Kassa, J. Zhang, D. Zheng, and D. Wei, "Short term wind power prediction using ANFIS," in *Proc. IEEE Int. Conf. power Renew. Energy (ICPRE)*, Oct. 2016, pp. 388–393.

[44] M. Fayaz, I. Ullah, and D. Kim, "An optimized fuzzy logic control model based on a strategy for the learning of membership functions in an indoor environment," *Electronics*, vol. 8, no. 2, p. 132, Jan. 2019.

[45] S. Ding, L. Guo, and Y. Hou, "Extreme learning machine with kernel model based on deep learning," *Neural Comput. Appl.*, vol. 28, no. 8, pp. 1975–1984, Aug. 2017.

[46] N.-D. Hoang and D. T. Bui, "Slope stability evaluation using radial basis function neural network, least squares support vector machines, and extreme learning machine," in *Handbook Neural Computation*. Amsterdam, The Netherlands: Elsevier, 2017, pp. 333–344.

[47] G.-B. Huang, "An insight into extreme learning machines: Random neurons, random features and kernels," *Cognit. Comput.*, vol. 6, no. 3, pp. 376–390, Sep. 2014.

[48] S.-J. Wang, H.-L. Chen, W.-J. Yan, Y.-H. Chen, and X. Fu, "Face recognition and micro-expression recognition based on discriminant tensor subspace analysis plus extreme learning machine," *Neural Process. Lett.*, vol. 39, no. 1, pp. 25–43, Feb. 2014.

[49] J. Wei, H. Liu, G. Yan, and F. Sun, "Robotic grasping recognition using multi-modal deep extreme learning machine," *Multidimensional Syst. Signal Process.*, vol. 28, no. 3, pp. 817–833, Jul. 2017.

[50] F. Wahid and D. H. Kim, "Short-term energy consumption prediction in Korean residential buildings using optimized multi-layer perceptron," *Kuwait J. Sci.*, vol. 44, no. 2, pp. 67–77, 2017.

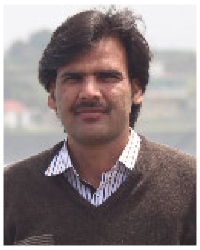
[51] F. Wahid and D. Kim, "A prediction approach for demand analysis of energy consumption using K-nearest neighbor in residential buildings," *Int. J. Smart Home*, vol. 10, no. 2, pp. 97–108, Feb. 2016.

[52] R. Penoyer, "The alpha-beta filter," *C User's J.*, vol. 11, no. 7, pp. 73–86, 1993.

[53] S. Rogers, "Alpha-beta filter with correlated measurement noise," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-23, no. 4, pp. 592–594, Jul. 1987.

[54] M. Vinaykumar and R. K. Jatoth, "Performance evaluation of alpha-beta and Kalman filter for object tracking," in *Proc. IEEE Int. Conf. Adv. Commun., Control Comput. Technol.*, May 2014, pp. 1369–1373.

JUNAID KHAN received the B.S. degree in computer science from the University of Swabi, Khyber Pakhtunkhwa, Pakistan, in 2017, and the M.S. degree in computer science from the Department of Computer Sciences, International Islamic University, Islamabad, Pakistan, in 2021. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea. In addition to his degree, he has completed short courses in machine learning, data science, and artificial intelligence. His research interests include intelligent algorithm, machine learning, deep learning, digital image processing, and data mining.



MUHAMMAD FAYAZ received the Ph.D. degree in computer engineering from Jeju National University (JNU), Jeju, South Korea, in 2019. During his studies, he worked with the Mobile Computing Laboratory, JNU. He is currently an Assistant Professor of computer science with the School of Arts and Sciences, University of Central Asia (UCA). Before joining UCA, he worked as a Visiting Lecturer and a Teaching Assistant with the University of Malakand, Chakdara, Pakistan, for a period of two years. His responsibilities included preparing project proposals, designing IoT-based smart solutions, the application and implementation of artificial intelligence algorithms, and writing project reports. During his stay in South Korea, he also worked on several projects on the underground system and the SAR-based big data classification and analysis of disaster/damage type, funded by the Electronics and Telecommunication Research Institute, and the Korean Aerospace Research Institute, respectively. Throughout his academic career, he has published over 60 research papers in the reputed Web of Science and Scopus indexed international journals and conferences. His research interests include combinatorial optimization problems, machine learning, image processing, the Internet of Things (IoT), fuzzy inference systems, and other related areas.

AYYAZ HUSSAIN (Member, IEEE) received the Ph.D. degree from the National University of Computer and Emerging Sciences NUCES-FAST, Islamabad, Pakistan, in 2009. He joined the Department of Computer Science, International Islamic University, Islamabad. He worked as a Postdoctoral Researcher with the Signal and Image Processing Laboratory, Gwangju Institute of Science and Technology, South Korea. He is currently working as an Associate Professor of computer science with Quaid-i-Azam University, Islamabad. His research interests include image processing, pattern recognition, machine learning, deep learning, and computational intelligence.

SHAH KHALID received the M.Sc. degree in computer science from the University of Peshawar, in 2004, the M.S. degree in computer network from Abasyn University, Peshawar, in 2012, and the Ph.D. degree in computer science from the University of Malakand, Chakdara, in 2018. He is currently an Assistant Professor of computer science with the University of Malakand, Pakistan. Before joining UOM, he worked as a Lecturer with the Government Degree College Lakari, Pakistan, for a period of two years. Throughout his academic career, he has published 15 research papers in well reputed journals and conference proceeding. His current research interests include computer graphics, virtual reality, collaborative computing, machine learning, networks, image processing, and other related areas.

WALI KHAN MASHWANI received the Ph.D. degree from Essex University, U.K., in 2011. He is currently working as an Associate Professor with the Institute of Numerical Sciences, Kohat University of Science and Technology (KUST). His research interests include evolutionary computation, single and multi-objective evolutionary optimization, decomposition-based evolutionary algorithms, artificial neural networks, game theory, numerical analysis, and mathematical programming and statistical models and its applications.

JEONGHWAN GWAK (Member, IEEE) received the Ph.D. degree in machine learning and artificial intelligence from the Gwangju Institute of Science and Technology, Gwangju, South Korea, in 2014. From 2002 to 2007, he had worked for several companies and research institutes as a researcher and a chief technician. From 2014 to 2016, he worked as a Postdoctoral Researcher with GIST, where he was a Research Professor, from 2016 to 2017. From 2017 to 2019, he was a Research Professor with the Department of Radiology, Biomedical Research Institute, Seoul National University Hospital, Seoul, South Korea. Since 2019, he has been an Assistant Professor with the Korea National University of Transportation (KNUT), where he is currently the Director of the Applied Machine Intelligence Laboratory. His current research interests include deep learning, computer vision, image and video processing, AIoT, evolutionary computation, optimization, and relevant applications of medical and visual surveillance systems.