Neuroscience Informatics 2 (2022) 100075

Contents lists available at ScienceDirect

Neuroscience Informatics

www.elsevier.com/locate/neuri

NaaS

Optimizing neural network based on cuckoo search and invasive weed optimization using extreme learning machine approach

Nilesh Rathod *, Sunil Wankhade

Mct's Rajiv Gandhi Institute of Technology Versova, Andheri (w), Mumbai 400053, India

A R T I C L E I N F O

Article history:
Received 24 March 2022
Accepted 19 April 2022

JEL classification:
C30
C45
C53
C60
C61
C69

Keywords:
Feed forward neural network
Extreme machine learning
Invasive weed optimization
Cuckoo search

A B S T R A C T

Extreme Learning Machine (ELM) is widely known to train feed forward network with high speed and good generalization performance. The only problem associated with ELM is required higher number of hidden neurons due to random selection. In this paper we proposed a new model Cuckoo Search with Invasive weed optimization based Extreme Learning Machine (CSIWO-ELM) to optimize input weight and hidden neurons. This model provides the optimize input to the feedforward network to improve the ELM. The developed model is experimented on three medical datasets to see the data classification. Also, the developed model is compared with different optimize algorithm. The experimental result proves the excellent working of CSIWO-ELM model for classification problem.

1. Introduction

One of the competent algorithms for training the single hidden layer feed-forward network is ELM [9]. In the network structure of ELM, the hidden layer's learning parameter is selected randomly. The least squares method helps in determining the hidden layer's output weight and the network offset and weight requires no iteration in this process. While considering the strong generalization ability and great training speed, the ELM algorithm is more favourable than the traditional neural network algorithm [6]. The fast-training speed is acquired mainly due to two points. The first point is the generation of latent layer parameters randomly and fixing the parameters without fine-tuning. The second point is the application of chain rule for the model parameter's partial derivative [10] and the anatomical derivation of output weights. In [11], the universal approximation capability is maintained by solving the output weights through the problems of regularized least squares. Thus, the ELM provided greater training speed and good generalization performance than the support vector machine and back-propagation based neural networks [12]. The traditional Support Vector Machine and ELM provide results that are biased to majority class by providing equal importance to all the samples. The problems regarding the class imbalance are solved using the ELM variants, such as Boosting WELM (BWELM) and Weighted ELM (WELM).

The imbalanced classification problems in real-life include the outnumbering of the sample that belongs to one class than other class. The minority class is the samples containing larger class proportion and the majority class is the samples having smaller proportion of class. The weights are initialized randomly within the hidden layer and input layer. The input data is mapped to feature space by allocating random weights [8]. The problems in imbalanced classification are solved using various classification methods. The classification methods are categorized into algorithmic level methods, cost sensitive methods and data level methods [18]. The class imbalance problems are reduced by altering the space in data using the data level methods, such as under-sampling and over-sampling [19,20]. Besides maintaining the balance in the distribution of data, the under-sampling method selects the portion of data from majority of class examples. The dataset is balanced by employing under-sampling with the help of Balance Cascade and Easy Ensemble algorithms [20]. Conversely, the synthetic minority class examples developed from informed oversampling method

* Corresponding author.
E-mail addresses: Nilesh.rathod@mctrgit.ac.in (N. Rathod), Sunil.wankhade@mctrgit.ac.in (S. Wankhade).

https://doi.org/10.1016/j.neuri.2022.100075

are balanced by class distribution, such as synthetic minority oversampling techniques (SMOTE) [21]. To address the learning of imbalanced data, the classifier design is modified using the algorithmic level methods directly [22,23]. With respect to examples of majority class, the minority class examples are misclassified by assigning more penalty in cost-sensitive methods [24,25,3].

The ELM algorithm based on efficient and simple single hidden layer FFNN learning algorithm is developed in [12,13]. These methods have greater training speed when compared with traditional methods. It also does not require iterative adjustments in the bias and weight of the network during the process of training. ELM-AE classification algorithm is an algorithm based on neural network in which the algorithm reproduced the auto encoder as well as input signal [14]. In kernel ELM-AE (KELM-AE), the KELM module is used as a two-layer base model. During the classification process, the second KELM method acts as a classification module and the completion matrix algorithm for the non-equilibrium labels is used in the label space [6]. However, the OC-ELM's shallow structure provided sub-optimal generalization performance by limiting the learning of dataset in high dimension and reducing the complex capability. In contrast, there is a fast development of ELM based on deep networks in the past years. The inspiration to the improvement of OC-ELMs is provided by the deep weighted ELM [15], kernel-based MK-ELM [17], stacked AEs based ML-ELM [14] and sparse representation based hierarchical ML-ELM (H-ELM) [16] for the representation of achievements and the investigation of multilayer neural network-based ELMs (ML-ELM) [4].

1.1. Review of literature

The existing ELM methods are described below, Eshtay et al. [1] developed a Competitive Swarm Optimizer (CSO) for data classification. This method had high stability and enhanced the generalization performance. However, the major issue lies in improving the training speed, network complexity, constancy and precision. Ertuğrul and Kaya [2] designed a data classification using Single layer ELM (sELM) and ELM based on linear regression (ELMr). Although this method provided effective solution, this method caused problems in overfitting. Shukla and Raghuwanshi [3] modelled an Online sequential class-specific extreme learning machine (OSCSELM) for classification of data. This method had low complexity in computation. However, this method failed to handle problems in imbalanced classification as it failed to consider the kernelized-based variant and multiclass-based variant of OSCSELM. Dai et al. [4] developed a data classification method using Multi-Layer-One Class Extreme Learning Machine (ML-OCELM). Although this method provided good generalization performance, this method failed to consider the cost function that was suitable for non-Gaussian noises and outliers.

Li et al. [5] designed a Evolutionary ELM for classification of data. This method had reduced consumption of time. However, this method had low performance as the adaptive weighting method with other classification algorithm was not implemented for imbalanced learning. Cheng et al. [6] modelled a Kernel ELM Auto Encoder algorithm for classification of data. Although this method had high accuracy in classification, this method failed to determine the relationship within the label and feature space along with the correlation of local label and feature selection. Cai et al. [7] designed a hybrid machine learning model for classification of data. This method effectively enhanced the ability of global search. However, this method failed to solve the problem of discrete optimization, such as scheduling of job and selection of feature. Raghuwanshi and Shukla [8] modelled a data classification method using Under Bagging ensemble method. Although this method had low cost of computation, the computational complexity during this process was high. But it has a wide variety of application like in smart vehicles, smart drainage system, speech recognition, material identification and so on [32–36].

1.2. Proposed hybrid intelligent approach

In this section, the proposed methodology adapted for classifying the data. The goal is to devise novel optimization driven extreme machine learning strategy with feed forward neural network for classification. The series of steps carried out for the classification are elaborated in this section. It is processed using three phases which involve Pre-Processing, Selection of Feature, and Classification of data. At first, the input data will be subjected to pre-processing phase for missing value imputation and transformation will be done using exponential kernel. Then, the feature selection process will be carried out using Jaro–Winkler distance for selecting the significant features for classifying the data. The feature selection is important in classification techniques for enhancing the performance. Choosing the features for classification poses numerous benefits, which involve data refining, minimizing the computational costs, and enhancing the classification accuracy. A good feature selection mechanism must have the capability to minimize the computational complexity and enhance the accuracy for classification. Thus, incorporating Jaro–Winkler distance in the process of feature selection offers some advantages for interpreting the models, as this measure provides some valuable features for the classification and can be utilized to handle the high dimensional data. Thus, the selected features will be subjected to the Feed forward neural network (FFNN) for the classification. The training algorithm will be newly developed by combining Extreme Learning Machine with hybrid optimization algorithm, namely CS-IWO. Here, the proposed CS-IWO will be designed by combining Cuckoo search (CS) [26] and Invasive Weed Optimization (IWO) algorithm [27]. Integration of CS in IWO will improve the convergence and thereby, provide an optimal solution, which will be considered here as the optimal solution for performing the classification. Here, multiple objectives will be developed with respect to the Loss function, Mean square error, entropy and accuracy. The proposed scheme will be implemented in Python tool. The dataset considered for the experimentation will be Heart Disease Dataset using Cleveland, Switzerland and Hungarian databases from the UCI machine learning repository [28,30,31]. The performance of the proposed scheme will be evaluated using three metrics, namely accuracy, sensitivity, and specificity. Moreover, a comparative analysis will be done by comparing the results attained with that of the work [1–3]. The schematic diagram of newly devised optimization algorithm namely Cuckoo search-based Invasive Weed Optimization (CS-IWO) for classifying data is depicted in Fig. 1.

2. System model

Data exploratory analysis revealed that the number of samples per source is very scarce. This is why we decided to collect every sample from each source and create a final database based on Cleveland, Hungary, and Switzerland. Thereafter, we realized there were many missing values (See Table 1). As we aim to perform machine learning algorithms, we were interested to have the most complete dataset.

2.1. Data imputation

Multivariate imputation has become one of the most appropriate methods to deal with missing data. We assume that the missing data are Missing At Random (MAR).

In other words, the probability that a value is missing depends only on observed values and not on unobserved values: "Any remaining missingness is completely at random". Here, we assume

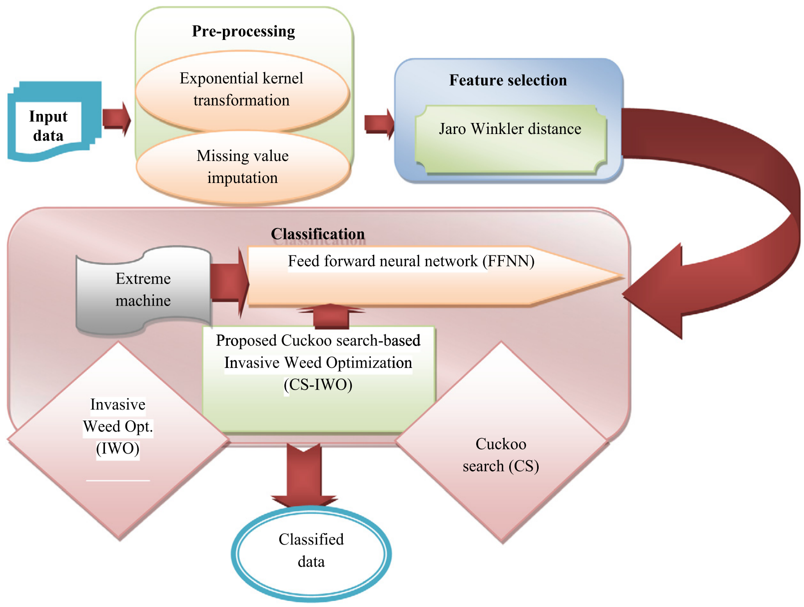


Fig. 1. Architecture diagram of developed CSIWO ELM-based FFNN.

Table 1
Percentage of missing values in features in each sample.

Features with missing values								
Dataset	Ca	Chol	Fbs	Oldpeak	Slope	Thal	Exang	trestbps
Cleveland	1.3%							
Switzerland	95.9%	100%	61%	4.9%	13..8%	42.3%	0.8%	
Hungarian	99%	7.8%	2.7%		64.6%	90.5%	0.3%	28%

the missing data is referred to as MAR type. Following on, we also had to clean out the new imputed values according to the structure of each feature. We are using Mean Substitution and regression substitution for missing value imputation.

2.2. Variable transformation

The dataset involved in this study is based on numerical and categorical data: a mix of data types. We have to transform these attributes to a suitable data format for the purpose of algorithm implementation.

Numerical attributes from any dataset may be measured in a different way (different units). Therefore, the features must be rescaled in order to have the same importance when applying any machine learning algorithm.

2.3. Exponential kernel transformation

The exponential kernel transformation [29] is used in pre-processing for transforming the data. The process of conversion of nominal features into numerical values is the data transformation. The exponential kernel function is given by the below expression as

$$K = \left(1 - \frac{\|P - R\|}{2d^2}\right)$$

Where, the smoothing factor and constant is represented by the term, d and R, respectively. The exponential kernel transformation improves the data classification process by improving the computation speed and simplifying the process of computation. The output data from the pre-processing process is denoted as P.

2.4. Transforming categorical data: one-hot-encoding (OHE)

We can distinguish between two types of categorical data: nominal and ordinal. The first type does not have any sense of order among discrete categorical values, while it does for ordinal data. In our dataset, we just have nominal data since there is no notion of order among the categorical values in any feature. The idea here is to transform the categorical features into a more representative numerical format which can be understood by the machine learning algorithms. Thus, first the categorical values should be transformed into numerical labels and then applying some encoding scheme to these values. Considering we have the numeric representation of any categorical attribute with m labels, the OHE scheme, encodes the feature into m binary attributes which can only contain a value of 0 (absence) or 1 (presence). For instance, if we have a categorical feature named chest pain type which contains 4 values: typical angina, atypical angina, non-anginal pain and asymptomatic. The first step will be to transform these values into numeric representation, and then generating 4 new features which would be cp1, cp2, cp3 and cp4 containing only 0 and 1 values in each new feature.

2.5. Feature selection

In several practical data mining situations, there are many attributes or features to be handled and most of them are clearly redundant or irrelevant. Many machine learning techniques try to select the most appropriate features, but this often leads to model performance deterioration. This can be improved by discarding those irrelevant attributes and keep the ones the models actually use. The advantages of feature selection are many. Reducing dimensionality speeds up the computation of those algorithms as well as providing a more compact and easy interpretable representation of the target. Moreover, it also reduces the problem of overfitting, where a learned machine learning model is tied up too closely to the training data. Therefore, it outperforms better on training data than on new unseen instances. In this study, we tried several feature selection approaches along with machine learning techniques to identify the most relevant attributes of the dataset. Attribute clustering can be useful for creating models. The idea behind hierarchical clustering is pretty simple: initially each attribute is considered as its own cluster. The algorithm then finds the two closest clusters in terms of similarity measure using Jaro Winkler, merge them and continue doing this until there is just one cluster left. A bottom-up approach hierarchical clustering that recursively merges features following the same basis as described previously. It uses the single linkage criterion which determines the distance (correlation) to use between sets of attributes. We can observe that some of the features are correlated with each other: (cp4 - exang1), (exang0 - thal ach) and (exang1 - old peak) are one of the sets of attributes with strongest correlation with each other.

Due to the imbalance of our dataset, we perform this algorithm with 10-fold stratified cross validation. Also for verification we used 5 fold cross validation. We used estimator, called Exponential kernel.

The number of features selected with this estimator, which gives the best score, is 10. Sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, ca.

Now consider the database used for obtaining the input data. The database is represented as

$$H = \{H_{mn}\}; (1 \le m \le A); (1 \le n \le B) \tag{1}$$

where the total data and the number of features is denoted as A and B respectively. The dataset is of the size $[A \times B]$, The m^{th} data in n^{th} feature is denoted as H_{mn}.

The exponential kernel function is given by the below expression as

$$K = \left(1 - \frac{\|P - R\|}{2d^2}\right) \tag{2}$$

where, the smoothing factor and constant is represented by the term, d and R, respectively. The exponential kernel transformation improves the data classification process by improving the computation speed and simplifying the process of computation. The output data from the pre-processing process is denoted as PH.

The pre-processing process is followed by the feature selection process in which the features are selected from the output data, obtained from the pre-processing process. In this research, the Jaro-Winkler distance is used for the efficient selection of the features. The Jaro-Winkler distance measures the similarity between the features for the efficient selection of features with the help of prefix factor. The similarity between the two features and Jaro-Winkler distance is expressed as

$$p = f(h_1, h_2) + (\beta \cdot \lambda (1 - f(h_1, h_2))) \tag{3}$$

$$f(h_1, h_2) = \begin{cases} 0 & ;\ \text{if } s = 0 \\ \frac{1}{3}\left(\frac{s}{|k_1|} + \frac{s}{|k_2|} + \frac{s-w}{s}\right) & ;\ \text{Otherwise} \end{cases} \tag{4}$$

where Jaro similarity metric and matching sequences are represented as f and s, respectively. The number of transposition and common prefix length is denoted as w and β, respectively. The length of two sequences is given as k_1 and k_2, respectively. The scaling factor is represented as λ. The suitable features for the data selection process are compared using the Jaro-Winkler distance. The output data obtained from the feature selection process is represented as $W = \{W_1, W_2, \ldots, W_l\}$. For further processing, the selected features are provided to the data classification module.

3. Methodology

The following steps describe the methodology of data classification on different dataset.

3.1. Data classification using developed CSIWO ELM-based FFNN

This section explains the developed CSIWO ELM-based FFNN method used for the classification of data. The FFNN [1] is trained using the CSIWO ELM for the classification of data. The input to the FFNN classifier is the output data obtained from the feature selection process. The developed CSIWO ELM method is the integration of ELM with hybrid optimization algorithm. The CSIWO is developed by integrating the CS algorithm in IWO. The CS algorithm helped in optimization of the hidden neurons and input weights of ELM. The pairwise competition within the particles is randomly selected from the swarm in the CS [26]. The winner of the competition is transferred directly to the next generation, whereas the loser of the competition is updated and transferred to the next generation. The CS provided better balance between exploitation and exploration besides solving the problems in high dimension. The IWO [27] depends on the colonization behaviour is invasive weed optimization. The IWO provided efficient solution against the different values of the parameter. Hence, the CS and IWO algorithms are integrated to provide better solution to the optimization problem. The architecture and pseudocode of FFNN are described below.

4. Problem formulation

The following pseudocode gives the brief about CSIWO ELM-based FFNN method which gives optimized solution.

Algorithm 1: Pseudocode of CSIWO ELM-based FFNN method.

Input: $\tau = 0, \eta, S(\tau)$
Output: Best M
Random initialization of $S(0)$
While $\tau <$ maximmumiteration
Determine the fitness function of all the individuals using equation $\text{Fitness} = \min((1 - L_F + A + MSE + \text{Entropy}))$
  Set the solution that fails to participate the competition, $O = S(\tau)$
$S(\tau + 1) = \beta$
while $O \ne \beta$ do
  Select two particles $M_1(\tau)$, $M_2(\tau)$ from O_{randomly}
  if *fitness of* $M_1(\tau) <$ *fitness of* $M_2(\tau)$ then
$M_w(\tau) = M_1(\tau)$ and $M_l(\tau) = M_2(\tau)$
  Else
$M_w(\tau) = M_2(\tau)$ and $M_l(\tau) = M_1(\tau)$
end if
For updating the value of $M_l(\tau + 1)$, the value of $M_l(\tau)$ is updated using equation $M_m^{(\tau+1)} = (\chi(\tau) - 1)/(\chi(\tau) - 2)[\mu \oplus \text{Levy}(\beta) - M_{\text{best}}/(\chi(\tau) - 1)]$
end while
$\tau = \tau + 1$
end while

Thus, the optimal solution provided by the CSIWO is used by the FFNN classifier for performing the classification.

4.1. Experimental setup

The developed CSIWO ELM-based FFNN method is implemented in Python tool. The input data for the developed CSIWO ELM-based FFNN method is extracted from the Heart Disease Dataset containing the Cleveland, Switzerland and Hungarian databases from UCI machine learning repository [28]. This dataset is a multivariate dataset. The number of attributes used in the database is 76 out of which the 14 are used. The presence of heart disease is described

Table 2
Comparative discussion of data classification methods (acc/sen/spe).

Dataset	Metrics	CSO-ELM	ELM-Sparse CM	KELM-AE	PSO-ELM	GA-ELM	Developed CSIWO ELM-based FFNN method
Cleveland	Sensitivity	0.7302	0.74269	0.8549	0.8701	0.8738	**0.8918**
	Accuracy	0.7611	0.7915	0.8864	0.8871	0.889	**0.7611**
	Specificity	0.8125	0.8248	0.8837	0.8907	0.893	**0.8125**
Switzerland	Sensitivity	0.7823	0.7870	0.9184	0.9237	0.9263	**0.9462**
	Accuracy	0.7901	0.8535	0.9233	0.9244	0.925	**0.9439**
	Specificity	0.851	0.8883	0.9203	0.9229	0.923	**0.9422**
Hungarian	Sensitivity	0.7901	0.8571	0.8994	0.9014	0.9045	**0.9248**
	Accuracy	0.823	0.8684	0.8989	0.901	0.9101	**0.823**
	Specificity	0.8559	0.878	0.9025	0.9057	0.9099	**0.8559**

Table 3
Comparative discussion of data classification methods (pre/rec/f1 on Cleveland).

Dataset	Metrics	Values	CSO-ELM	ELM-Sparse CM	KELM-AE	PSO-ELM	GA-ELM	Developed CSIWO ELM-based FFNN method
Cleveland	Precision	0	0.69	0.75	0.758	0.806	0.862	0.929
		1	0.75	0.818	0.714	0.778	0.846	0.923
		2	0.692	0.643	0.733	0.786	0.833	0.923
		3	0.75	0.667	0.75	0.8	0.667	0.833
		4	0.333	0.5	1	1	1	1
	Recall	0	0.69	0.724	0.862	0.862	0.862	0.897
		1	0.75	0.75	0.417	0.583	0.917	1
		2	0.692	0.692	0.846	0.846	0.769	0.923
		3	0.6	0.8	0.6	0.8	0.8	1
		4	0.5	0.5	1	1	0.5	0.5
	F1-Score	0	0.69	0.737	0.806	0.833	0.862	0.912
		1	0.75	0.783	0.526	0.667	0.88	0.96
		2	0.692	0.667	0.786	0.815	0.8	0.923
		3	0.667	0.727	0.667	0.8	0.727	0.909
		4	0.4	0.5	1	1	0.667	0.667

using the integer from 0 to 4. The number of attributes present in developed CSIWO ELM-based FFNN method is 303.

4.2. Evaluation metrics

The evaluation metrics used for determining the performance developed CSIWO ELM-based FFNN method are specificity, accuracy and sensitivity.

Sensitivity: Specificity is the measure of actual positives that are identified correctly in the developed CSIWO ELM-based FFNN method.

Specificity: Specificity is the measure of actual negatives that are identified correctly in the developed CSIWO ELM-based FFNN method.

Precision: Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances.

Recall: Recall (also known as sensitivity) is the fraction of relevant instances that were retrieved. Both precision and recall are therefore based on relevance.

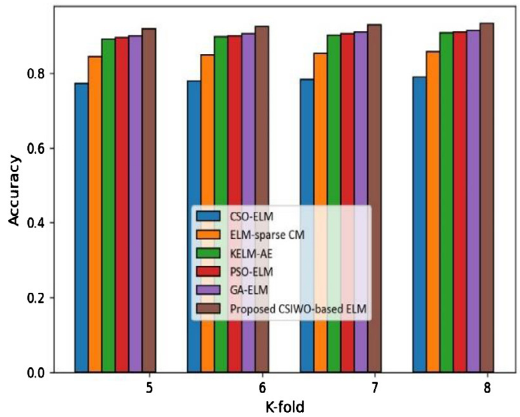
F1-Score: The F-score, also called the F1-score, is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into 'positive' or 'negative'.

5. Result analysis

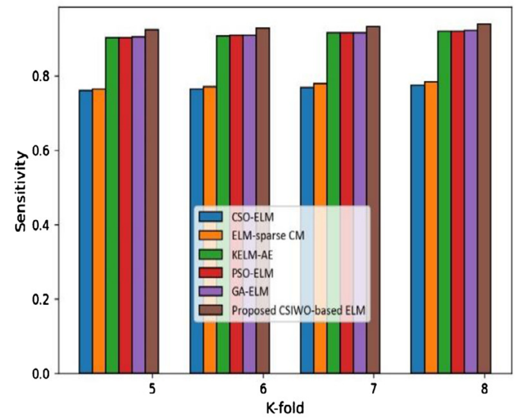
Table 2 shows the comparative discussion of data classification methods using the metrics for the percentage of training data of 90. For the training data of 90%, the developed CSIWO ELM-based FFNN method has a maximum sensitivity of 0.8918, whereas the existing CSO-ELM, ELM-Sparse CM, KELM-AE, PSO-ELM and GA-ELM methods have the sensitivity of 0.7302, 0.74269, 0.8549, 0.8701 and 0.8738, respectively in Cleveland dataset. On comparing with the existing CSO-ELM, ELM-Sparse CM, KELM-AE, PSO-ELM and GA-ELM methods accuracy of 0.7494, 0.7916, 0.8864, the developed method obtained a maximum accuracy of 0.7611 for the training data of 90% in Cleveland dataset. The developed CSIWO ELM-based FFNN method has a maximum specificity of 0.8125 and the existing CSO-ELM, ELM-Sparse CM, KELM-AE, PSO-ELM and GA-ELM methods have the specificity of 0.8125, 0.8248, 0.8837, 0.8907 and 0.893 for the training data of 90% in Cleveland dataset.

On comparing with the existing CSO-ELM, ELM-Sparse CM, KELM-AE, PSO-ELM and GA-ELM methods sensitivity of 0.7823, 0.7870, 0.9184, 0.9237 and 0.9263, the developed method obtained a maximum sensitivity of 0.9462 for the training data of 90% in Switzerland dataset. The developed CSIWO ELM-based FFNN method has a maximum accuracy of 0.9439 and the existing CSO-ELM, ELM-Sparse CM, KELM-AE, PSO-ELM and GA-ELM methods have the accuracy of 0.7901, 0.8535, 0.9233, 0.9244 and 0.925 for the training data of 90% in Switzerland dataset. For the training data of 90%, the developed CSIWO ELM-based FFNN method has a maximum specificity of 0.9422, whereas the existing CSO-ELM, ELM-Sparse CM, KELM-AE, PSO-ELM and GA-ELM methods have the specificity of 0.851, 0.8883, 0.9203, 0.9229 and 0.923, respectively in Switzerland dataset.

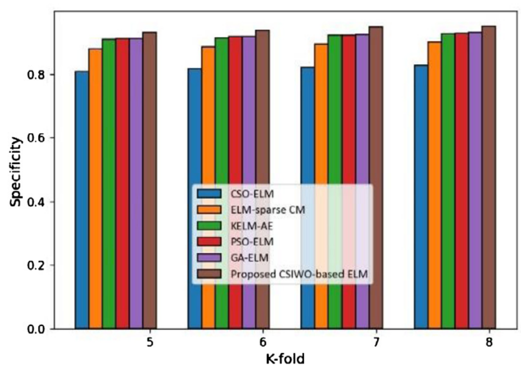
The developed CSIWO ELM-based FFNN method has a maximum sensitivity of 0.9248 and the existing CSO-ELM, ELM-Sparse CM, KELM-AE, PSO-ELM and GA-ELM methods have the sensitivity of 0.7901, 0.8571, 0.8994, 0.9014 and 0.9045 for the training data of 90% in Hungarian dataset. For the training data of 90%, the developed CSIWO ELM-based FFNN method has a maximum accuracy of 0.823, whereas the existing CSO-ELM, ELM-Sparse CM,



K fold Result on Switzerland w.r.t ***Accuracy***



K fold Cross Result on Switzerland w.r.t ***Sensitivity***



K fold Result on Switzerland w.r.t ***Specificity***

Fig. 2. K fold Validation graph on Switzerland Dataset.

Table 4
Comparative discussion of data classification methods (pre/rec/f1 on Switzerland).

Dataset	Metrics	Values	CSO-ELM	ELM-Sparse CM	KELM-AE	PSO-ELM	GA-ELM	Developed CSIWO ELM-based FFNN method
Switzerland	Precision	0	1	1	1	1	1	1
		1	0.7	0.75	0.833	0.833	0.833	1
		2	0.25	0.333	0.5	0.5	0.667	1
		3	0.7	0.778	0.889	0.889	0.875	0.833
	Recall	0	0.5	0.5	1	1	1	1
		1	0.636	0.818	0.909	0.909	0.909	0.818
		2	0.5	0.5	0.5	0.5	1	1
		3	0.7	0.7	0.8	0.8	0.7	1
	F1-Score	0	0.667	0.667	1	1	1	1
		1	0.667	0.783	0.87	0.87	0.87	0.9
		2	0.333	0.4	0.5	0.5	0.8	1
		3	0.7	0.737	0.842	0.842	0.778	0.909

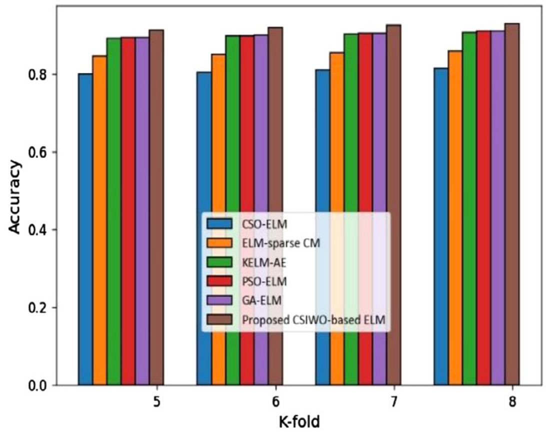
KELM-AE, PSO-ELM and GA-ELM methods have the accuracy of 0.823, 0.8684, 0.8989, 0.901 and 0.9101, respectively in Hungarian dataset. On comparing with the existing CSO-ELM, ELM-Sparse CM, KELM-AE, PSO-ELM and GA-ELM methods specificity of 0.8559, 0.878, 0.9025, 0.9057 and 0.9099, the developed method obtained a maximum specificity of 0.8559 for the training data of 90% in Hungarian dataset.

Table 3 gives the comparative discussion on precision recall and f1 score on Cleveland dataset. As compared to GA-ELM and other optimised technique developed CSIWO ELM based FFNN method provide good result in terms of defined performance parameters. The developed method gives precision 0.929 at 0, recall 1 and recall 0.96 at level 1. Similarly in Tables 4 and 5 the effective results are shown for the developed model.

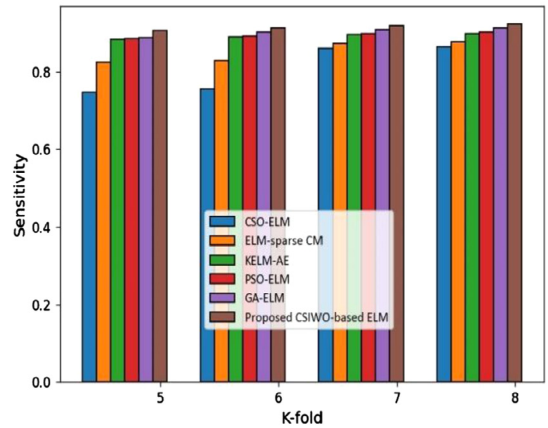
Table 5 shows the experiment analysis related with the accuracy, specificity, sensitivity, precision, recall and f1-score. From the result in Table 5 we can analyse that our proposed csiwo-elm model performs very well. The results are compared with the other model i.e. cso-elm, elm-sparse cm, kelm-ae, pso-elm, ga-elm. While observing the result we can say that our model not only performs for improving the evaluation parameters but also shows the model stability.

5.1. K fold validation result

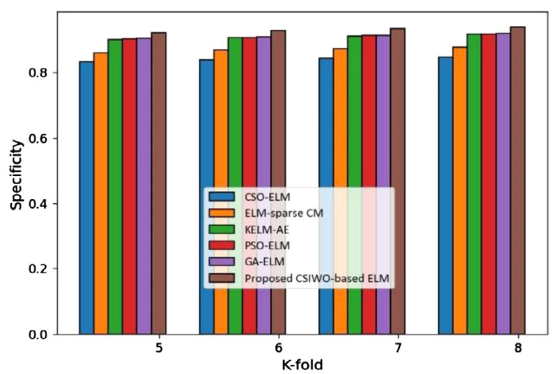
The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a



K fold Result on Hungarian w.r.t ***Accuracy***



K fold Result on Hungarian w.r.t ***Sensitivity***



K fold Result on Hungarian w.r.t ***Specificity***

Fig. 3. K fold Validation graph on Hungarian Dataset.

Table 5
Comparative discussion of data classification methods (pre/rec/f1 on Hungarian).

Dataset	Metrics	Values	CSO-ELM	ELM-Sparse CM	KELM-AE	PSO-ELM	GA-ELM	Developed CSIWO ELM-based FFNN method
Hungarian	Precision	0	0.889	0.968	0.969	0.97	0.97	0.944
		1	0.5	0.75	0.6	1	1	0.75
		2	0.667	0.615	0.727	0.727	0.727	1
		3	0.571	0.857	0.857	0.714	0.714	0.875
		4	0.333	0.75	0.75	0.75	0.75	1
	Recall	0	0.889	0.833	0.861	0.889	0.889	0.944
		1	0.5	0.75	0.75	1	1	0.75
		2	0.667	0.889	0.889	0.889	0.889	0.889
		3	0.571	0.857	0.857	0.714	0.714	1
		4	0.333	1	1	1	1	1
	F1-Score	0	0.889	0.896	0.912	0.928	0.928	0.944
		1	0.5	0.75	0.667	1	1	0.75
		2	0.667	0.727	0.8	0.8	0.8	0.941
		3	0.571	0.857	0.857	0.714	0.714	0.933
		4	0.333	0.857	0.857	0.857	0.857	1

specific value for k is chosen, it may be used in place of k in the reference to the model, such as k = 10 becoming 10-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

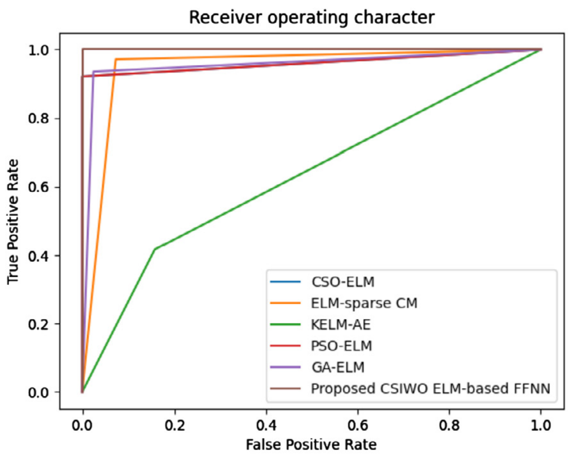
Figs. 2 and 3 show the K-fold validation of Switzerland and Hungarian dataset w.r.t. Accuracy, Sensitivity and Specificity.

5.2. ROC curve

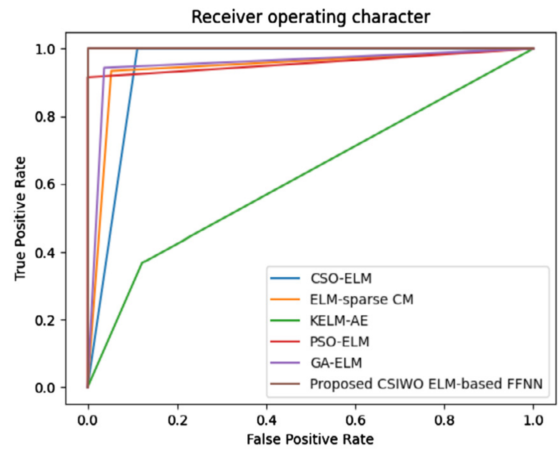
Fig. 4 shows the ROC graph on Hungarian dataset.

6. Discussion

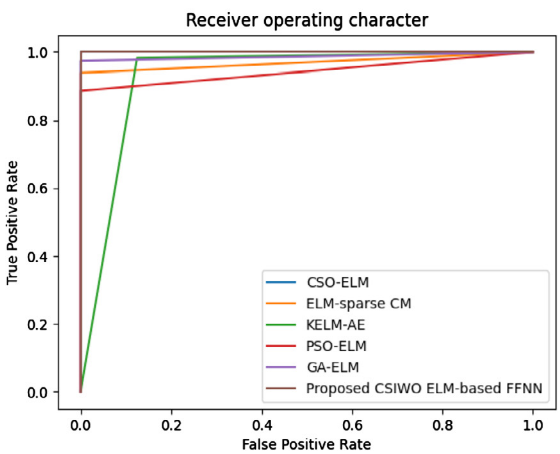
By the above features engineering steps, we have selected the best suitable features. And these features effectively send to the CSIWO-ELM model. We can see the effective results once the model is properly structured. So, by comparing CSIWO-ELM with



ROC graph for Cleveland dataset



ROC graph for Hungarian dataset



ROC graph for Switzerland dataset

Fig. 4. ROC graph on Hungarian Dataset.

different optimised algorithm we can see that our model obtained better classification accuracy. Also, it is minutely observed that CSIWO model achieved better specificity and sensitivity. This gives the model authenticity for the result.

Another interesting observation is precision, recall and f1 score which gives the better model stability. CSIWO model provides reliable results as compared with other optimised algorithms. By looking at K-fold chart and ROC chart they provide us brief analysis on the how significantly different our CSIWO-ELM model with other optimized model. The chart shows the stability of the CSIWO-ELM model even if higher number of neurons.

7. Conclusion

This paper presented an approach based on Cuckoo Search with invasive weed optimisation and extreme learning machine for training feed forward network. The proposed model CSIWO-ELM will optimise input weights and hidden neurons. The analysis is done using 3 datasets which are Cleveland, Hungarian, and Switzerland. Our proposed model is compared with CSO-ELM, ELM-Sparse CM, KELM-AE, PSO-ELM, GA-ELM. The experiment result shows that CSIWO not only performed better than other algorithms by recording different parameters but also reduce the training time. CSIWO-ELM provide From the experiment result we can conclued that CSIWO model provides the stability in the model.

8. Future work

Here we have provided intelligent hybrid approach for CSIWO – ELM model but in future we can individually modify the cuckoo search algorithm and invasive weed optimization algorithm to get improved classification accuracy. Also, we can perform different model stability test. Medical analysis in neurological function is also a great advantage to get the better accuracy in order to get the result [37,38].

Human and animal rights

The authors declare that the work described has not involved experimentation on humans or animals.

Declaration of competing interest

The authors declare that they have no known competing financial or personal relationships that could be viewed as influencing the work reported in this paper.

Funding

This work did not receive any grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author contributions

All authors attest that they meet the current International Committee of Medical Journal Editors (ICMJE) criteria for Authorship.

References

[1] M. Eshtay, H. Faris, N. Obeid, Improving extreme learning machine by competitive swarm optimization and its application for medical diagnosis problems, Expert Syst. Appl. 104 (2018) 134–152.
[2] Ö.F. Ertuğrul, Y. Kaya, A detailed analysis on extreme learning machine and novel approaches based on ELM, Am. J. Comput. Sci. Eng. 1 (5) (2014) 43–50.
[3] S. Shukla, B.S. Raghuwanshi, Online sequential class-specific extreme learning machine for binary imbalanced learning, Neural Netw. 119 (November 2019) 235–248;
H. Dai, J. Cao, T. Wang, M. Deng, Z. Yang, Multilayer one-class extreme learning machine, Neural Netw. 115 (2019) 11–22.
[4] H. Li, X. Yang, Y. Li, L.Y. Hao, T.L. Zhang, Evolutionary extreme learning machine with sparse cost matrix for imbalanced learning, ISA Trans. 100 (2020) 198–209.
[5] Y. Cheng, D. Zhao, Y. Wang, G. Pei, Multi-label learning with kernel extreme learning machine autoencoder, Knowl.-Based Syst. 178 (2019), pp. 1-0.
[6] Z. Cai, J. Gu, J. Luo, Q. Zhang, H. Chen, Z. Pan, Y. Li, C. Li, Evolving an optimal kernel extreme learning machine by using an enhanced grey wolf optimization strategy, Expert Syst. Appl. 138 (2019) 112814.
[7] B.S. Raghuwanshi, S. Shukla, Class imbalance learning using UnderBagging based kernelized extreme learning machine, Neurocomputing 329 (2019) 172–187.
[8] G-B. Huang, Q-Y. Zhu, C-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501;
P.J. Werbos, Generalization of backpropagation with application to a recurrent gas market model, Neural Netw. 1 (4) (1988) 339–356.
[9] G.B. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, Neural Netw. 17 (4) (2006) 879–892.
[10] G.B. Huang, An insight into extreme learning machines: random neurons, random features and kernels, Cogn. Comput. 6 (3) (2014) 376–390.
[11] G. Huang, G.B. Huang, S. Song, K. You, Trends in extreme learning machines: a review, Neural Netw. 61 (2015) 32–48.
[12] L.L. Kasun, H. Zhou, G.B. Huang, C.M. Vong, Representational learning with extreme learning machine for big data, IEEE Intell. Syst. 28 (6) (2013) 31–34.
[13] T. Wang, J. Cao, X. Lai, B. Chen, Deep weighted extreme learning machine, Cogn. Comput. 10 (6) (2018) 890–907.
[14] J. Tang, C. Deng, G. Huang, Extreme learning machine for multilayer perceptron, IEEE Trans. Neural Netw. Learn. Syst. 27 (4) (2016) 809–821.
[15] C.M. Wong, C.M. Vong, P.K. Wong, J. Cao, Kernel-based multilayer extreme learning machines for representation learning, IEEE Trans. Neural Netw. Learn. Syst. 29 (3) (2016) 757–762.
[16] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging, boosting and hybrid-based approaches, IEEE Trans. Syst. Man Cybern., Part C, Appl. Rev. 42 (2012) 463–484.
[17] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (2009) 1263–1284.
[18] X.Y. Liu, J. Wu, Z.H. Zhou, Exploratory undersampling for class imbalance learning, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 39 (2009) 539–550.
[19] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[20] D.A. Cieslak, T.R. Hoens, N.V. Chawla, W.P. Kegelmeyer, Hellinger distance decision trees are robust and skew-insensitive, Data Min. Knowl. Discov. 24 (2012) 136–158.
[21] Y. Tang, Y. Zhang, N.V. Chawla, S. Krasser, SVMs modeling for highly imbalanced classification, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 39 (2009) 281–288.
[22] J. Mathew, C.K. Pang, M. Luo, W.H. Leong, Classification of imbalanced data by oversampling in kernel space of support vector machines, IEEE Trans. Neural Netw. Learn. Syst. (2018) 1–12.
[23] W. Zong, G.-B. Huang, Y. Chen, Weighted extreme learning machine for imbalance learning, Neurocomputing 101 (2013) 229–242.
[24] X.S. Yang, S. Deb, Engineering optimisation by cuckoo search, Int. J. Math. Model. Numer. Optim. 1 (4) (2010) 330–343.
[25] S. Karimkashi, A.A. Kishk, Invasive weed optimization and its features in electromagnetics, IEEE Trans. Antennas Propag. 58 (4) (2010) 1269–1278, Heart Disease Data Set taken from https://archive.ics.uci.edu/ml/datasets/heart+disease, 2020.
[26] Dingding Yang, Liming Tang, Shiqiang Chen, Jun Li, Image segmentation based on exponential kernel function, in: 14th International Symposium on Pervasive Systems, Algorithms and Networks, 2017, pp. 293–298.
[27] W. Cai, J. Yang, Y. Yu, Y. Song, T. Zhou, J. Qin, PSO-ELM: a hybrid learning model for short-term traffic flow forecasting, IEEE Access 8 (2020) 6505–6514.
[28] G.S. Krishnan, S. Kamath, A novel GA-ELM model for patient-specific mortality prediction over large-scale lab event data, Appl. Soft Comput. 80 (2019) 525–533.
[29] Nilesh Rathod Sunil Wankhade, Review of optimization in improving extreme learning machine, EAI Endorsed Trans. Ind. Netw. Intell. Syst. 8 (28) (2021), https://doi.org/10.4108/eai.17-9-2021.170960.
[30] S. Gaikwad, S. Patel, A. Shetty, Brain tumor detection: an application based on machine learning, in: 2021 2nd International Conference for Emerging Technology (INCET), 2021, pp. 1–4.
[31] UCI heart disease data set retrieved from http://archive.ics.uci.edu/ml/datasets/heart+disease, September 26, 2018. (Accessed 20 June 2018).
[32] Ashish Uplenchwar, Modern speech identification model using acoustic neural approach, Int. J. New Pract. Manag. Eng. 6 (03) (2017) 1–6, https://doi.org/10.17762/ijnpme.v6i03.58.
[33] D.N. Jain, Artificial neural network models for material classification by photon scattering analysis, Int. J. New Pract. Manag. Eng. 9 (03) (2020) 1–4, https://doi.org/10.17762/ijnpme.v9i03.86.
[34] B. Maruthi Shankar, Neural network based hurdle avoidance system for smart vehicles, Int. J. New Pract. Manag. Eng. 8 (04) (2019) 1–7, https://doi.org/10.17762/ijnpme.v8i04.79.
[35] Rahul Sharma, Monitoring of drainage system in urban using device free localization neural networks and cloud computing, Int. J. New Pract. Manag. Eng. 7 (04) (2018) 8–14, https://doi.org/10.17762/ijnpme.v7i04.69.
[36] Naveen Jain, Artificial neural network models for material classification by photon scattering analysis, Int. J. New Pract. Manag. Eng. 9 (03) (2020) 1–4, https://doi.org/10.17762/ijnpme.v9i03.88.
[37] G. Roland, S. Kumaraperumal, S. Kumar, A.D. Gupta, S. Afzal, M. Suryakumar, PCA (principal component analysis) approach towards identifying the factors determining the medication behavior of Indian patients: an empirical study, Tob. Regul. Sci. 7 (6) (2021) 7391–7401.
[38] N.R. Nayak, S. Kumar, D. Gupta, A. Suri, M. Naved, M. Soni, Network mining techniques to analyze the risk of the occupational accident via Bayesian network, Int. J. Syst. Assur. Eng. Manag. (2022) 1–9.