Received August 21, 2019, accepted August 26, 2019, date of publication August 29, 2019, date of current version September 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2938271

A Novel Evolutionary Algorithm for Data Classification Problem With Extreme Learning Machines

ENDER SEVINC
Computer Engineering Department, University of Turkish Aeronautical Association, 06790 Ankara, Turkey
e-mail: esevinc@thk.edu.tr

ABSTRACT Machine learning techniques have gained great popularity due to their success in data classification problems. This study proposes a novel evolutionary feature selection algorithm integrated with Single Hidden Layer Feed-forward Neural Networks (SLFN)s. Our main goal is to find out the most efficient subset of features and provide the best prediction accuracy. The algorithm combines the evolutionary technique of genetic algorithms (GA) and calculates the fitness values (prediction accuracy) of each selected subset of features by using Extreme Learning Machines (ELM). The results of the SLFN are calculated in a faster manner, which is very suitable for the GA while optimizing the selection of the best subset of features. The experimental results show that the proposed algorithm provides significant improvements. Competitive results are obtained/verified by comparing our solutions with those of the state-of-the-art data classification algorithms.

INDEX TERMS Extreme learning machine, feature selection, genetic algorithm, SLFN.

I. INTRODUCTION

Feature selection methodologies and techniques have attracted the interest of many scientists. This interest seems to continue since there is not a discovered exact feasible solution yet. In this study, we propose a new algorithm for Single hidden Layer Feed-forward Neural network (SLFN)s. This research is believed to put forward a fast and accurate way for predicting a reasonable learning level for a SLFN, which is a linear system where information always moves in one direction. Generalized inverse operation of the hidden layer output matrix is used to determine the output weights of the links between the hidden and the output layers. The solution process starts by choosing random values for the input weights and the hidden layer biases. Based on this concept, the ELM comes up with a very fast learning capacity compared to traditional learning methods and reaches better generalization performance especially on SLFNs. The ELM is also known to get the smallest training error [1].

Researchers have studied and put forward many feature selection methods in the literature. These methods generally fall into three main groups: filter [2], wrapper [3], [4] and embedded/hybrid [5]. In the filter method, features are ranked and ordered with respect to some predefined measures. However, the results might be inefficient for improving the performance of the learning algorithms since the selection of features is an independent process and might have a negative effect on the prediction accuracy. The wrapper methodologies adopt a strategy to explore the combinatorial space of feature selection in order to train the network. They evaluate the usefulness of features based on the performance of the classifier. The wrapper algorithms perform better but consume a lot of time compared to filter methods. Finally, the embedded/hybrid methods are similar to wrappers but computationally less expensive. They select features by using a specific model. Decision tree learning methods, such as ID3 and C4.5 are the examples of these methods [5].

Despite many available data, machine learning techniques still lack interpreter systems that have desirable accuracy levels. For example, there might be large amounts of data at hand however, deciding the type and the phase of the disease of a patient might be quite erroneous or misleading. It is always hard to select the best set of features because the total search space is intractable. Besides this, the prioritization of the features is another area to concentrate on since it is quite

The associate editor coordinating the review of this article and approving it for publication was Xiangtao Li.

difficult and the success rate of the learning may be lower than expected.

We put forward a novel evolutionary wrapper method, integrating ELM and GA. With the help of this integration, the learning capacity of SLFN is improved remarkably by discovering and using the most suitable subset of features. Briefly, learning is performed by ELM that is introduced by Huang [1], [6], [7]. Its high speed and accuracy for reaching results are integrated with the evolutionary GA techniques. With the help of GA, the existing best results are improved. Competitive results with the state-of-the-art algorithms are presented in the experimental section.

Section 2 discusses the literature of data classification problem. The proposed feature subset selection algorithm and its ability to achieve good learning results are explained in Section 3. Section 4 shows the improvement gained by the proposed algorithm based on accuracy and learning level, comparison with recent state-of-the-art algorithms is discussed. Finally, conclusions and possible future works are discussed in the last section.

II. RELATED WORKS

There are many studies in the literature and we will discuss most recent state-of-the-art supervised machine learning techniques for the solution of the data classification problem. Related works about GA and ELM are also provided in this section.

The filter, wrapper and hybrid methods are the mainly used techniques for the feature selection problem. One of the initial studies in the literature is a wrapper feature subset selection method implemented by using supervised learning methods [8]. However, many of the studies ignore the role of features and try to put forward a way to improve only the prediction accuracy. In another study [9], the solution is grouped into sub-parts with respect to the number of attributes for local management. A segmented crossover operator and a segmented mutation operator are put forward in order to operate on these segments. The aim is to avoid invalid chromosomes. Similarly, the study in [10] examines possible strategies to improve the efficiency while generating the initial population. Some other studies use filter methods to have more information about features for better prediction values. The study in [11] concentrate on mutual information, and measures the amount of information that the feature subset S contains about the output classes C. The proposed algorithm tries to find out pattern classification based on mutual information and a mutual information between the predictive labels of a trained classifier and the true classes.

Many studies employ various processes for optimizing the parameters of the classifiers. Study in [12] encodes two parameters of Support Vector Machine (SVM) and the feature subsets into a chromosome. In [13] researchers try to optimize the input feature subset selection and the parameters of SVM by using a particle swarm optimization (PSO) algorithm. The proposed model is hybrid one in which PSO and SVM algorithms are integrated for improving the classification accuracy on a small and appropriate feature subset. However, some parameters of SVM need to be optimized repeatedly. Because of such defects, SVM is commonly preferred as the main classifier in many studies such as Naive Bayes algorithm, logistic regression, and C4.5 decision trees. SVM's good prediction accuracy and good generalization ability is commonly known and confirmed. One important drawback of SVM is that it has a relatively slow learning speed. Especially, when the SVM is compared to ELM, it needs more time to execute and this problem becomes more evident as the data set size gets larger.

Besides wrapper and filter methods, embedded methods (hybrid methods) are examples of recent and attractive topics for feature selection. They are reported to perform feature selection by the help of a training process, which is managed by a learning machine [2]. Decision tree learners, such as ID3 [14] and C4.5 [5] are examples for embedded methodologies. Another alternative, the recursive feature elimination approach being a recently proposed feature selection algorithm is based on the SVM theory and shows a good performance on the problems of gene selection for the micro array data. [15]. In a similar approach for neural networks, instant parameter prediction models are also presented in neural network-based wireless environments as presented in [16].

There are multi-objective feature selection methods in neural networks as well. They aim more than one goal at the same time and taking attention. One of them in [17] is a clustering problem for patient stratification. The proposed algorithm tries to remove irrelevant, redundant, and noisy features concurrently. Another similar multi-objective algorithm is proposed in [18] and it tries to balance the exploration and the stochastic exploitation capability for reaching a better solution. In [19], the proposed algorithm exploits the strength of the discrete biogeography based optimization for the classification method and tries to find the smaller feature subsets in order to get rid of irrelevant genes in data set. In another study in [20], the proposed algorithm has a multi-objective goal for ranking binary classification by the help of artificial bee colony algorithm. It also uses ELM in order to select the most important feature that can maximize the sensitivity and specificity while ignoring redundant and noisy features.

A. GENETIC ALGORITHM

GA is a well-known population based optimization algorithm. It starts with a set of random solutions (population) and searches for a better solution through generations. In each generation, newer solutions are produced by using crossover and mutation operators. When the GA is run for a sufficient amount of time/iterations, it will be able to obtain good solutions that will be close (or the same) to the optimum value. After GA uses crossover and mutation operators to calculate new solutions, the algorithm evaluates the fitness value of each solution separately. This fitness value is used in the following steps for selecting individuals that will form the next generation.

GA process begins with a set of random individuals (population). Each individual is a solution representation. An individual is characterized by a set of parameters known as genes that form a chromosome. The fitness function determines the performance of the solutions. This function gives a numeric value. The probability that an individual is going to be selected for reproduction is based on its fitness value. GA reaches the optimum solution using an evolutionary process. It uses the crossover and mutation operators with a selection mechanism. The selection process is used for choosing the best individuals and pass their genes to the next generation. The crossover is a well-known operator of the most significant phases of a GA. For each pair of individuals to be mated, a crossover point is chosen at random for the chromosomes and an offspring is created by combining the genes within the cut points of the chromosome. Finally, the mutation operator randomly selects one or more genes with a low random probability and changes it. Mutations simply change the value of a gene [21].

Pairs of solutions are randomly selected and mated, and they produce new ones with operators. After finding the fitness values of all new solutions, the population is sorted with respect to fitness values and the next generation is executed. This process goes on until the algorithm converges to a point (value). After reaching this point, we decide on the best solution in the population, which is the chromosome having the best-fitness value [22].

Finally, finding the best subset of features is known to be an NP-hard problem. An exhaustive evaluation of possible feature subsets is not feasible and the application of GA for feature selection can be suitable as listed below: firstly, they are more capable of avoiding getting stuck in local optima and secondly, they may be classified as a standard methodology that can generate the best subsets any time and improve the quality of selected features.

B. EXTREME LEARNING MACHINE

The ELM proposes a learning methodology for the SLFNs. This technique is intensively used because of its extreme speed when compared to that of traditional feed-forward network learning algorithms. The ELM is commonly used for deriving learning methods in many areas because of its low computational complexity and accuracy [23].

The output of SLFN having L number of hidden nodes can be represented with (1) below;

$$f_L(x) = \sum_{i=1}^{L} \beta_i . G(a_i, b_i, x) \quad x \epsilon R^n, \; a_i, b_i \epsilon R \tag{1}$$

where a_i and b_i are learning parameters of hidden nodes and β_i is the weight connecting the i^{th} hidden node to the output node. $G(a_i, b_i, x)$ is the output of the i^{th} hidden node with respect to the input x. [1], [6], [7].

In a SLFN with L number of hidden nodes with activation function $g(x)$, being able to approximate L samples with zero error means that activation function g(x) can approximate these L samples with zero error is equal to $\sum_{j=1}^{L} \|o_j - t_j\| = 0$.

This means that there exists β_i, a_i and b_i in (2) such that;

$$\sum_{i=1}^{L} \beta_i . g(a_i . x_j + b_i) = t_j \quad j = 1, \ldots, N \tag{2}$$

If we rewrite this equation in another way as given in (3) for better understanding;

$$H\beta = T \tag{3}$$

where,

$$H(a_1, \ldots, a_L, b_1, \ldots, b_L, x_1, \ldots, x_N) = \begin{bmatrix} g(a_1 . x_1 + b_1) & \cdots & g(a_L . x_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(a_1 . x_N + b_1) & \cdots & g(a_L . x_N + b_L) \end{bmatrix}_{NxL} \tag{4}$$

and,

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{Lxm} \quad and \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{Nxm} \tag{5}$$

β^T is the transpose of a matrix or vector β and H is called the hidden layer output matrix of the network in (4) and (5). The i^{th} column of H is the i^{th} hidden node's output vector with respect to inputs $x_1, x_2, \ldots, x_N$ and the j^{th} row of H is the output vector of the hidden layer with respect to input x_j.

For fixed input weights a_i and the hidden layer biases b_i, to train an SLFN is simply equivalent to finding the least-squares solution $\hat{\beta}$ of the linear system in (3).

Since in most cases the number of hidden nodes *(L)* is much less than the number of distinct training samples and the smallest norm least squares solution of the linear system is as in

$$\hat{\beta} = H^{\dagger} T \tag{6}$$

where $H^{\dagger}$ is the Moore-Penrose (MP) generalized inverse of matrix H. [24]

III. THE PROPOSED ALGORITHM (FS-ELM)

As initially introduced by the studies [1], [6], [7], the ELM can obtain acceptable solutions within extremely shorter periods. In this study, we aim to use this capability by integrating it with a GA. Thus with the help of this integration, the results can be evolved through iterations. The best practices of these two algorithms/methods are combined into our proposed algorithm named as ***Feature Selection with ELM (FS-ELM)***. The main goal of FS-ELM is to have a powerful and fast method for the classification of the data for binary and multi-classes in a reasonable period.

Each chromosome is a solution with a set of selected features. This selected set of features constitutes a network, i.e. SLFN, and then it is solved by the ELM. The fitness of a network is evaluated by dividing the number of instances that are predicted correctly to the total number of instances in the test data set, as presented in (7)

$$Fitness(s) = \frac{\#\,Instances\ predicted\ correctly}{\#\,Total\ Instances} \tag{7}$$

A chromosome shows a sequence of genes and F denoting *feature*, is the sequence of the inputs in the original data set file. The place of a feature, F is important and put to the same place as it is in the original file. This is shown in detail in the next sections.

A. CHROMOSOME STRUCTURE

The selection process is explained by a sample data set with 8 features as given in Fig. 1. If the i^{th} feature (F_i) is selected then $F_i = 1$, otherwise $F_i = 0$ and it is not selected. This sequence constitutes a solution and presents the chromosome structure which is used for this study.

A sample chromosome structure is presented in Fig.1.

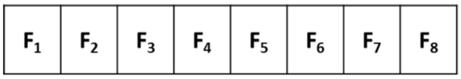

FIGURE 1. Chromosome structure.

The gene sequence forms a sample chromosome structure. Each $F_1..F_8$ denotes the gene in its place in the original data set file. Consequently, the genes denote the features in the data set file.

TABLE 1. Parameters of the *GA*.

Parameter	Value
Initial Population Size	200
Convergence ratio	95%
Crossover type	truncate, 2-point
Truncate ratio	50%
Crossover ratio	0.6 (60%)
Mutation ratio	0.01 (1%)

Crossover operator is used for mating the chromosomes. We use two-point crossover with parameters as in Table 1. In the experiments, we obtain better results with two-point crossover when compared to that of one-point. In this method, two points are selected on parent solutions. The segments between these two cut points are swapped between these parents. Then *Offspring-1* is formed as seen in Fig. 2. Each parent, *P1* or *P2*, is samples chromosome as in Fig. 1. Then within the same procedure *Offspring-2* is produced similarly.

Mutation operator is similar to its commonly known form. A gene is selected according to the mutation probability that is 1% as presented in Table1. The number of genes is equal to the number of features. The selected gene is flipped (0/1) in the mutation process.

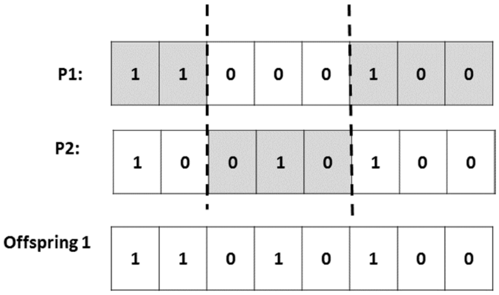

FIGURE 2. Crossover operation.

Selection is performed among the most elite chromosomes. After sorting the chromosomes with respect to their fitness values, the population is divided into two equal parts. The selection operator works in the upper part of the population data set. This elite part is again divided into two equal parts and then parents are selected respectively starting from the top of each sub-part. After offspring creation, its fitness is calculated. Then after the resorting process, the worse part equals to the half of the population size which is truncated as stated in Table 1.

Table 2 shows the data sets used in the experiments. The data sets are obtained from the UCI data set repository [25]. These are the data sets used by most of the state-of-the-art studies in the literature. The instance numbers, attribute/feature numbers, and output class numbers are presented in Table 2. Data sets will be referred with their *"ID"*s in the study from now on.

TABLE 2. Data set descriptions.

Data set	ID	# instances	# attributes	# classes
Vehicle	VEH	846	18	4
WDBC	WDB	569	32	2
Ionosphere	ION	351	34	2
Chess	CHS	3196	36	4
Sonar	SON	208	60	2
Musk	MUS	168	476	2
Pima-Indian Diabetes	PID	768	8	2
Wisconsin Breast Cancer(Original)	WIS	699	10	2
Waveform	WAV	569	21	3
Spambase	SPM	4601	57	2

The number of hidden neurons *(L)* is an important parameter in SLFNs. There have been numerous analyses to use the appropriate L for a higher learning rate. This value in this study is meant to be proportional to the input size, however, it is selected as 30 at least and 60 at most, as in most of the studies. The pseudo code of the FS-ELM is presented in Algorithm 1.

B. PHASES OF FS-ELM

There are two phases, namely *GA* and *ELM* in the proposed algorithm. In the *GA* phase, an initial population is generated and all these solutions are evaluated by the ELM. In this phase, input weights and hidden layer bias matrices

Algorithm 1 FS-ELM Algorithm

```
Input: instance m,
        size n of population,
        rate e of elitism,
        k of iterations
Output: Solution X

Begin
Create initial population with n random solutions
Evaluate the fitness values of all random solutions
Sort the population with respect to fitness values
for (i = 1 to k) do
    Select best individuals w.r.t. rate e of elitism;
    Generate new offspring using new operators;
    Evaluate the fitness of new individuals;
    while (New Individuals are present) do
        Randomly generate input weights and biases;
        Calculate hidden-layer output matrix H;
        Calculate (β and T matrices);
        Evaluate the fitness of new offspring(inst. m);
    Re-sort the population
    Truncate the worst individuals in the population
//loop will end up due to a stopping criterion
End
```

of each solution are randomly assigned and formed [1]. After executing these matrices with the activation function *g(x)*, Moore-Penrose inverse of H matrix is obtained [24]. H matrix is used for calculating the output weight β matrix in the next calculations. Then, the minimum norm least-squares solution of the system is solved, and the output classes are predicted in the final phase. These are all explained as given in (2) - (6) in Section 2.

After calculating the fitness of the solutions using the ELM, chromosomes in the population are sorted with respect to their fitness values. New offspring is created by using the crossover and mutation operators. Calculations are performed in iterations/generations. Each iteration produces a new generation in which each new offspring is evaluated and inserted into the population. Then the results converge to a point that the population of solutions can not be improved any further. After k number of iterations as given in Algorithm 1, the FS-ELM reaches to its convergence point. It means the algorithm terminates its iterations and the best solution is put forward.

In SLFNs, hidden neurons are modeled using an activation function for the output classes. This function is used to learn and understand the functional mappings between input and output node points for neural networks. These functions transform the input signal to the output node. In the experiments, *Sigmoid* function is used as the activation function since it is known to be one of the most successful ones. Sigmoid is a non-linear, monotonic and S-shaped activation function that produces a value in the range [0, 1]. In this context, sigmoid function is a special form of logistic function and is defined below;

$$sig(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

Sigmoid has a vanishing gradient problem which occurs because of multiplying many small numbers to compute gradients of the ''front'' layers in a neural network. Though having such a problem, *Sigmoid* is more commonly used and accepted to achieve better learning rates with respect to other activation functions. The rest of the parameters are as in Table 1 for GA and ELM part of FS-ELM algorithm.

C. LEARNING METHODOLOGY

A sample flowchart of the FS-ELM algorithm is depicted in Fig. 3. We *train and test* sessions respectively for calculating the fitness value of a solution. These training and testing processes are all executed with 10-fold cross-validation methodology. This technique is used to remove the effect of random data selection processes. Then the average result of testing phases is assigned as the fitness value of that solution. Cross-validation technique is a statistical method used to predict the performance of machine learning methodologies. It is commonly used in applied machine learning to compare and select a model for a given predictive modelling problem. 10-fold cross-validation is commonly used and known, which means the ***''k''*** value of k-fold cross-validation is taken as ''10'' throughout this study. For implementing the FS-ELM, each data set is divided into 10 equal parts, the first 9 pieces are used for learning and the last part is used for testing. In this way, all parts are subjected to the same process sequentially,

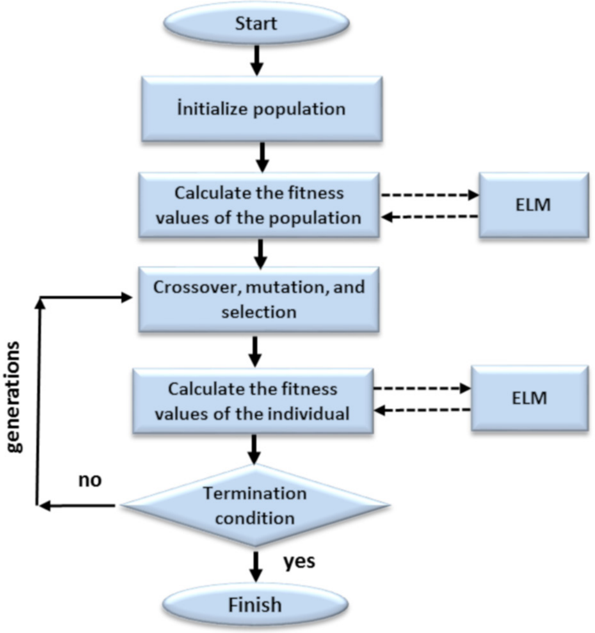
Start
İnitialize population
Calculate the fitness values of the population
ELM
Crossover, mutation, and selection
Calculate the fitness values of the individual
ELM
generations
no
Termination condition
yes
Finish

FIGURE 3. FS-ELM Algorithm flowchart.

and due to 10 folds, all sub-parts are rotated 10 times in the same way. Any fitness value of a solution is assigned due to the result of 10-fold cross-validation process in FS-ELM.

In the testing phase, we predict the output nodes according to the linear system defined in (3). This output is important and supplies the fitness of the current SLFN with respect to the features selected. Then that value is given as the learning rate of that network/solution.

In the following rounds of 10-fold cross-validation, we take the other 9 sub-parts, one by one, from the original file, and create the training and testing data sets from scratch. Thus, every one of the 10 sub-parts takes place 9 times in training data set and one time in the testing data set.

In the final phase given in Fig. 3, all rounds are calculated and then the average of all turns are evaluated. Each chromosome is a distinct solution for the SLFN and the fitness value of that chromosome/solution is decided and used in the following iterations.

IV. EXPERIMENTAL SETUP

The experiments are carried out on a PC having i5-4200U 1.60 GHz CPU with 8 GB of RAM (Windows 7-64-bit). The FS-ELM algorithm is coded by using Java language and tested also with MATLAB (v. R2016a).

A. THE PERFORMANCE OF THE FS-ELM ALGORITHM

Table 3 gives the results of experiments with the FS-ELM. According to initial results, the *feature selection with FS-ELM* is beneficial with respect to the case that *All features are included*. In *"All features selected"* case, the fitness values are again evaluated with ELM. However, feature selection by FS-ELM eliminates some features and evolves. As a result, this process has a positive effect on the learning rate of the network when compared to that of all features.

The proposed algorithm outperforms *"All features selected"* case as given in Table 3. This is because of dirty data and useless features in the files that degrade the performance. Therefore, if we can get rid of those features degrading the learning, we can improve the prediction accuracy performance.

Additionally, FS-ELM is observed to increase the prediction accuracy remarkably for not only binary but also multi-class data sets. Numerically, the increase in multi-class prediction accuracy is better than that of binary classes. VEH and CHS are sample multi-class files having higher accuracy rates for prediction than overall average of the total as seen in Table 3.

If we consider the data sets in Table 3, it can be observed that 33.2% performance increase is achieved in the average. This is believed to be a remarkable effect of the proposed algorithm. If the results for such data sets are closely examined, the effect of FS-ELM can be seen more precisely. For example, CHS having 3196 instances with 36 attributes, WDB having 569 instances with 32 attributes, SPM having 4601 instances with 57 attributes, and WAV having 569 instances with 21 attributes are good examples.

TABLE 3. Performance of FS-ELM due to all features included.

ID	FS-ELM selects features	All features selected	Improvement (%)
VEH	0.8384	0.2482	59.0
WDB	0.9771	0.5010	47.6
ION	0.9342	0.5075	42.7
CHS	0.9753	0.5019	47.3
SON	0.8809	0.6300	25.1
MUS	0.7795	0.5638	21.6
PID	0.7787	0.5561	22.3
WIS	0.9766	0.9360	4.1
WAV	0.7128	0.3942	31.9
SPM	0.9072	0.6028	30.4
		Average:	33.2

In Table 3, we can see that the performance increase is more than the average of CHS, WDB, SPM, and WAV data sets. This shows that the performance of the FS-ELM is better especially with medium/big sized data sets. Most probably, these data sets might have more irrelevant and noisy data that degrade the learning rate. Besides that, the benefit of FS-ELM is seen not only in accuracy but also in the execution times as well.

A detailed view of the features is provided in Table 4, *"ID", "Accuracy", "Selected features" and "Execution Time (s)"* columns are shown while "Accuracy" column is the same as the "FS-ELM selects feature subset" column of Table 3. The values presented in this column are decimal values between 0.0 and 1.0, which denote 0% and 100% of accuracy levels respectively. The third column, namely "Selected features" of Table 4 shows a gene sequence having 0's and 1's inside. The genes in the sequence are separated by "–"s and it straightforwardly shows that if the value of that feature is "1", the gene in that sequence/place in the original file will be selected, if it is "0" then it will not. The gene in that position is the feature stated in the same position as in the original data set file.

The last gene in the third column denotes the output class. This gene/feature is used for testing during the execution of training processes. As a result, the last columns are always "1" in Table 4. Throughout this paper, they are included as if it is one of the features. Because when you check the data sets, that feature presenting the output classes is included in the original data file and takes place as the first or the last feature of data sets. When you visit the related website in ELM website [23], the feature number of the data sets is declared as if the output class is also a feature. In order to examine the execution way of FS-ELM, a sample data set will be selected, namely Breast Cancer Wisconsin (Original) file. This data set is named as WIS as in Table 2. WIS is a relatively smaller data set and easier to explain. WIS data set has 9 features and 1 column more for the output classes, adding up to 10 features totally. This 9-feature data set can be found with the same name and structure in UCI website [25].

After executing the FS-ELM algorithm for WIS, we get the result, "1-1-0-0-0-1-1-0-0-1" as given in the third column of

TABLE 4. Selected features by FS-ELM algorithm.

ID	Accuracy(%)	Selected features	Exe.Time (s)
VEH	0.8384	1-1-0-1-1-1-0-1-0-1-0-0-1-1-1-0-1-1-1	2145
WDB	0.9771	1-1-0-0-1-0-1-0-1-0-0-0-0-0-0-1-1-0-0-1-1-0-0-1-0-1-1-1-0-1	2691
ION	0.9342	0-1-1-0-1-0-0-1-0-0-0-0-0-0-0-0-0-0-1-0-0-0-0-0-1-0-0-0-0-0-0-0-1	2365
CHS	0.9753	1-0-0-1-0-1-0-0-0-1-0-0-0-1-1-0-1-0-0-0-1-0-0-0-0-0-0-0-0-0-0-1-1-0-1-0-1	16189
SON	0.8809	0-0-0-0-0-1-0-0-0-0-1-0-0-0-0-0-1-0-0-0-0-0-0-0-0-0-0-1-0-1-0-1-0-0-0-0-1-0-0-0-0-1-1-0-0-0-0-0-1-1-0-0-1-0-1-0-1-0-0-0-1	1646
MUS	0.7795	0-0-0-0-1-1-0-1-1-0-0-0-0-0-0-1-1-1-0-1-0-0-0-1-1-1-1-1-0-0-0-0-1-1-0-1-1-1-0-0-1-0-0-1-0-1-1-0-0-0-0-0-0-1-1-0-0-0-0-1-0-0-1-0-1-0-1-0-0-0-0-1-0-0-0-0-0-1-0-1-0-0-0-0-1-0-0-1-0-0-0-0-0-0-1-0-0-1-0-0-0-1-0-0-1-0-0-0-1-0-0-0-1-0-0-1-1-1-1-0-0-0-1-1-0-1-0-1-1-0-1-0-0-0-0-1-1-0-0-1-0-1-1-0-1-0-1-0-1-0-1-0-1-0-0-0-1-1-1-0-0-1-1-0-1-0-1	3839
PID	0.7787	0-1-0-0-0-1-0-1-1	591
WIS	0.9766	1-1-0-0-0-1-1-0-0-1	338
WAV	0.7128	0-0-0-1-1-0-1-0-0-0-0-0-0-0-0-0-1-1-1-0-0-0-0-0-0-0-0-0-0-0-1-0-0-0-0-0-0-0-0-0-0-1	18271
SPM	0.9072	1-0-0-1-1-0-1-0-0-0-0-0-0-1-0-1-0-0-0-0-0-0-1-1-1-1-1-0-0-1-1-0-1-0-1-0-0-1-1-1-0-1-0-1-1-1-0-1-0-0-1-1-1-0-0-0-0-1	12277

Table 4. Totally 10 columns are mapped in which the final column shows the output class. Therefore, we focus on the first 9 columns, i.e. "1-1-0-0-0-1-1-0-0" for the WIS data set. This sequence means that 1^{st}, 2^{nd}, 6^{th} and 7^{th} columns are selected and the rest are ignored.

According to the result presented here, if we consider only these 4 attributes rather than the whole set, we obtain 97.7% prediction accuracy value. If the same process is repeated by using all of the features, we can only achieve 93.6% fitness as given in Table 3.

The last column of Table 4 gives the execution time of the algorithms. It can be observed that ELM is enormously fast during the solution of the problem. After calculating fitness values, new offspring are created with FS-ELM operators. This process is repeated within the most elite group of chromosomes in the population and then the population is sorted and truncated. The solution population gradually converges to an accuracy value. Then after k-fold cross-validation procedures have been executed, the best found values are assigned as the fitness value of related solutions, which are presented in Table 4.

CHS, SPM and WAV data sets having relatively higher execution times are important. If we examine these data sets, they have many instances and features which can degrade the accuracy. FS-ELM is believed to handle such kind of data sets successfully in terms of accuracy and execution time.

B. COMPARISON WITH STATE-OF-THE-ART ALGORITHMS

The first study as one of the state-of-the-art algorithms is a recent study presented in [26] and quite similar to our study in terms of using GA and ELM for feature selection. The proposed algorithm is named as "hybrid GA and ELM based feature selection algorithm (HGEFS)". This algorithm is mainly a wrapper feature selection method and provides comparison results with many other similar methodologies. The compared methodologies are as given below;

- four filter methods: Correlation-based Feature Subset selection (CFS), ReliefF, Gain Ratio and ChiSquare,
- two-hybrid wrapper methods: particle swarm optimization-support vector machines (PSO-SVM) and GA-ELM,
- two embedded methods: C4.5 and SVM-RFE,
- three ensemble feature selection methods: Attribute Bagging (AB), Multi-View Adaboost (MVA) and Random Subspacing Ensemble (RSE).

With the help of this study, the search space will be expanded at least to the scope of the study in [26] and a fair comparison with all of the state-of-the-art algorithms will be presented. As a result, the values of all of these methods will be shown and compared with our study results. Another recent study is in Kiziloz *et al.* [27] and proposes a multi-objective Teaching Learning Based Optimization (TLBO) algorithm. It is a wrapper method and works on feature selection by utilizing its algorithm-specific parameters in order to improve the speed or accuracy. Multi-objective TLBO with Scalar Transformation (MTLBO-ST), Multi-objective TLBO with Non-Dominated Selection (MTLBO-NS) and Multi-objective TLBO with Minimum Distance (MTLBO-MD) are the variants of their main algorithm in [27].

They claim that MTLBO-NS achieves higher prediction accuracy values for the same number of features and provides multi objective solutions with higher accuracy values spending more amount of time. The results of MTLBO-NS are presented in the comparison table since it is one of the most effective classification algorithms.

One of the common and highlighted points of these studies is that they use the same data sets from UCI repository [25]. Although some data sets are not presented in all studies, the others are presented in our study. Besides that, PID data set is reported to be excluded from [25] currently. However, it can be found easily on the Internet [28].

Since there are too many results to present using one table, the results will be separated into two tables. Similar approaches are given in the same table. Table 5 shows the results of mostly filter and wrapper methods while Table 6 presents embedded, ensemble, and hybrid methods.

In Tables 5 and 6 some entries are marked as "N/A" since those data sets are *"Not Applicable"*. For example in [26], PID and WIS data sets are not included in the experiments. Besides that, there are some unused data sets currently missing in the [25] website. Additionally, some of the data sets have undefined data inside, e.g. "Arrhythmia" data set. If that data set is checked, some input values are seen as "?". This which means *"not mentioned" or "not defined"* which is

TABLE 5. Comparison with filter and wrapper algorithms.

ID	Unselected [26]	ReliefF [26]	Gain Ratio [26]	ChiSquare [26]	CFS-SFS [26]	C 4.5 [26]	HGEFS [26]	MTLBO-NS [27]	FS-ELM
VEH	0.7881	0.8087	0.7948	0.7976	0.6917	0.7364	0.8202	N/A	**0.8384**
WDB	0.9446	0.9554	0.9446	0.9511	0.9580	0.9314	0.9710	N/A	**0.9771**
ION	0.8779	0.8757	0.8763	0.8786	0.8906	0.9116	0.9133	0.8900	**0.9342**
CHS	0.9120	0.9643	0.9435	0.9398	0.9403	**0.9943**	0.9874	N/A	0.9753
SON	0.7750	0.7870	0.7945	0.7765	0.7875	0.7115	0.8300	N/A	**0.8809**
MUS	0.7774	0.7781	0.7872	0.7863	0.7955	0.8487	0.8813	**0.9260**	0.7995
PID	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.7710	**0.7787**
WIS	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.9750	**0.9766**
Average	0.8458	0.8615	0.8568	0.8550	0.8439	0.8557	0.9005	0.8905	**0.9009**

TABLE 6. Comparison with other embedded/ensemble/hybrid algorithms.

ID	SVM-RFE [26]	PSO-SVM [26]	GA-ELM [26]	AB [26]	MVA [26]	RSE [26]	HGEFS [26]	FS-ELM
VEH	0.7849	0.8076	0.8168	0.8095	0.8120	0.7732	0.8202	**0.8384**
WDB	0.9262	0.9468	0.9653	0.9514	0.9573	0.9484	0.9710	**0.9771**
ION	0.8984	0.9012	0.9036	0.8954	0.9014	0.8901	0.9133	**0.9342**
CHS	0.9778	0.9682	0.9763	0.9512	0.9537	0.9486	**0.9874**	0.9753
SON	0.8073	0.8128	0.8216	0.8023	0.8017	0.7950	0.8300	**0.8809**
MUS	0.8542	0.8496	0.8626	0.8527	0.8563	0.8473	**0.8813**	0.7995
Average	0.8748	0.8810	0.8910	0.8771	0.8804	0.8671	0.9005	**0.9009**

unacceptable for any supervised algorithm. It means there is no data for that input of instance. On the other hand, there is no clue how the study in [26] managed such cases. All the attributes in a SLFN must be defined with a value; otherwise, neural network cannot be mapped from input to output nodes. As a result, such kind of data sets are skipped and excluded in the experimental comparisons of our study.

Because of these problems, 8 of 10 data sets can be used from [26] and [27] in Table 5 and 6 of 10 data sets can be included in Table 6. The results for each data set are given in rows and the bold values in that row show the best result.

In the experiments, it can be noticed that the proposed algorithm, FS-ELM can deal, manage and perform better than other state-of-the-art algorithms. Although FS-ELM is not the best for all the data rows, it performs better than the others mostly. It is also noticed that there is no available result especially for huge data sets such as SPM and WAV. However, it can be seen that FS-ELM performs better than HGEFS in [26] and MTLBO-NS in [27] for data sets such as CHS and VEH. These are relatively bigger data sets among others. The average given for FS-ELM is the average value for the first 6 data sets since the rival algorithm (HGEFS in [26]) of FS-ELM has only the results of 6 data sets. For being fair, the average of these are given in Table 5.

The results are also similar for other embedded, ensemble, and hybrid algorithms presented in Table 6. FS-ELM performs better than HGEFS and the other stated algorithms in [26]. Though FS-ELM is not the best for all cases, it produces better results mostly.

The execution times of the algorithms in [26] are reported as for C4.5 and SVM-RFE in Weka platform are 0.6 and 226.4 seconds, respectively. The processing times of AB, MVA and RSE are 5.3, 6.4 and 4.7 seconds, respectively and for the wrapper methods, the processing times of PSO-SVM, GA-ELM and HGEFS are 50493.1, 4373.8 and 4936.7 seconds, respectively in [26]. Similarly, the execution time of MTLBO-NS in [27] is reported as 2988, 2399, 223, 352 seconds for ION, MUS, PID, WIS respectively. FS-ELM is a GA with a time complexity of $O(m.n)$ where m is the number of samples and n is the number of features as in HGEFS of [26]. Execution times of FS-ELM are given in Table 4 due to related data sets. It can be seen that we have similar execution times with HGEFS and MTLBO-NS. On the other hand, HGEFS values are given by using Weka platform for [26] and the execution time may change due to the termination criterion in any GA for us.

Finally, it can be concluded that our proposed algorithm is better than the recent state-of-the-art algorithms in a reasonable amount of execution time and produces more competitive results. Especially, the performance is remarkable when huge data sets are classified.

V. CONCLUSION

In this paper, a new evolutionary GA with ELM is proposed. This study focuses on a wrapper feature selection algorithm that predicts and forms a network to map the input nodes to its output counterparts. The proposed algorithm works uniquely to reach the best solutions. One important result of this study is that not all of the features are needed for predicting a better output. The experiments yield 33.2% performance increase in the average between selecting the best subset of features and all features. The redundant/noisy degrades the learning rate of the SLFN. The results obtained by the FS-ELM are better in 6 out of 8 data sets from the UCI repository when compared with state-of-the-art algorithms. The structure and distribution of data play a key role in the performance of the algorithm. We can claim that the FS-ELM algorithm can be used for having a reasonable learning rate in SLFNs.

Feature selection has attracted great attention and we believe that this study can be used as a reference for other future research in this field. The parallel computation power of GPU can be a promising area for new and better studies. Another interesting future work can be on decision of the number of hidden neurons of a neural network. Because it is noticed that the learning rate of the network and the time of execution is strongly dependent on the number of hidden neurons.

REFERENCES

[1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

[2] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[3] A. Torralba and A. Oliva, "Depth estimation from image structure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1226–1238, Sep. 2002. doi: 10.1109/TPAMI.2002.1033214.

[4] M. A. Hall and L. A. Smith, "Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper," in *Proc. FLAIRS Conf.*, May 1999, pp. 235–239.

[5] J. R. Quinlan, "Improved Use of Continuous Attributes in C4.5," *J. Artif. Intell. Res.*, vol. 4, pp. 77–90, Mar. 1996.

[6] G.-B. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, pp. 155–163, Dec. 2010.

[7] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[8] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, 1997.

[9] W. Yang, D. Li, and L. Zhu, "An improved genetic algorithm for optimal feature subset selection from multi-character feature set," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2733–2740, 2011. doi: 10.1016/j.eswa.2010.08.063.

[10] X. Li, N. Xiao, C. Claramunt, and H. Lin, "Initialization strategies to enhancing the performance of genetic algorithms for the *p*-median problem," *Comput. Ind. Eng.*, vol. 61, no. 4, pp. 1024–1034, 2011.

[11] J. Huang, Y. Cai, and X. Xu, "A hybrid genetic algorithm for feature selection wrapper based on mutual information," *Pattern Recognit. Lett.*, vol. 28, no. 13, pp. 1825–1844, May 2007.

[12] C.-L. Huang and C.-J. Wang, "A GA-based feature selection and parameters optimizationfor support vector machines," *Expert Syst. With Appl.*, vol. 31, no. 2, pp. 231–240, 2006.

[13] C.-L. Huang and J.-F. Dun, "A distributed PSO-SVM hybrid system with feature selection and parameter," *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1381–1391, 2008.

[14] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[15] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.

[16] A. Akbas, H. U. Yildiz, A. M. Ozbayoglu, and B. Tavli, "Neural network based instant parameter prediction for wireless sensor network optimization models," *Wireless Netw.*, vol. 25, no. 6, pp. 3405–3418, 2019.

[17] X. Li and K.-C. Wong, "Multiobjective patient stratification using evolutionary multiobjective optimization," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 5, pp. 1619–1629, Sep. 2018.

[18] X. Li and M. Yin, "Multiobjective binary biogeography based optimization for feature selection using gene expression data," *IEEE Trans. Nanobiosci.*, vol. 12, no. 4, pp. 343–353, Dec. 2013.

[19] B. Liu, M. Tian, C. Zhang, and X. Li, "Discrete biogeography based optimization for feature selection in molecular signatures," *Mol. Informat.*, vol. 34, no. 4, pp. 197–215, 2015.

[20] X. Li, M. Li, and M. Yin, "Multiobjective ranking binary artificial bee colony for gene selection problems using microarray datasets," *IEEE/CAA J. Automatica Sinica*, to be published.

[21] E. Sevinc and M. Karakaya, "Planning multiple UAVs to visit points of interest considering flight range and service time constraints," in *Proc. Int. Conf. Eng. Natural Sci. (ICENS)*, Sarajevo, Bosnia and Herzegovina, May 2016, pp. 26–28.[Online]. Available: https://www.icens.eu/album/icens-2016

[22] E. Sevinc and T. Dökeroğlu, "A novel hybrid teaching-learning-based optimization algorithm for the classification of data by using extreme learning machines," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 27, no. 2, pp. 1523–1533, 2019.

[23] *MATLAB Codes ELM Algorithm*. Accessed on: Jul. 19, 2019. [Online]. Available: https://www.ntu.edu.sg/home/egbhuang/elm_random_hidden_nodes.html,

[24] G. Feng, Z. Qian, and X. Zhang, "Evolutionary selection extreme learning machine optimization for regression," *Soft Computing*, vol. 16, 9, pp. 1485–1491, 2012.

[25] *UCI Mach. Learn.Repository*. Accessed on: Jul. 19, 2019. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.php

[26] X. Xue, M. Yao, and Z. Wu, "A novel ensemble-based wrapper method for feature selection using extreme learning machine and genetic algorithm," *Knowl. Inf. Syst.*, vol. 57, no. 2, pp. 389–412, Nov. 2018.

[27] H. E. Kiziloz, A. Deniz, T. Dokeroglu, and A. Cosar, "Novel multiobjective TLBO algorithms for the feature subset selection problem," *Neurocomputing*, vol. 306, Sep. 2018, pp. 94–107. doi: 10.1016/j.neucom.2018.04.020.

[28] *Pima Indians Diabetes Database*. Accessed on: Jul. 19, 2019. [Online]. Available: https://www.kaggle.com/uciml/pima-indians-diabetes-database

ENDER SEVINC was born in Istanbul, Turkey, in 1969. He received the B.S. degree from Electrical/Electronical Department, Military Academy, and the M.S. and Ph.D. degrees from Computer Engineering Department, Middle East Technical University, Ankara, in 2000 and 2009, respectively.

From 2014 to 2017, he was an Engineer with IT industry. Since 2017, he has been an Assistant Professor with the Computer Engineering Department, University of Turkish Aeronautical Association. He is the author of ten articles and five conference proceedings. His research interests include optimization, machine learning, and genetic algorithms.