IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 11, 2019, accepted December 27, 2019, date of publication January 8, 2020, date of current version January 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964803

Genetic Algorithm for the Mutual Information-Based Feature Selection in Univariate Time Series Data

UMAIR F. SIDDIQI[1], **(Member, IEEE), SADIQ M. SAIT**[1,2], **(Senior Member, IEEE), AND OKYAY KAYNAK**[3], **(Fellow, IEEE)**

[1]Center for Communications and IT Research, Research Institute, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[2]Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[3]Department of Electrical and Electronic Engineering, Bogazici University, Istanbul 80815, Turkey

Corresponding author: Sadiq M. Sait (sadiq@kfupm.edu.sa)

This work was supported by the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

ABSTRACT Filters are the fastest among the different types of feature selection methods. They employ metrics from information theory, such as mutual information (MI), Joint-MI (JMI), and minimal redundancy and maximal relevance (mRMR). The determination of the optimal feature selection set is an NP-hard problem. This work proposes the engineering of the Genetic Algorithm (GA) in which the fitness of solutions consists of two terms. The first is a feature selection metric such as MI, JMI, and mRMR, and the second term is the overlapping-coefficient that accounts for the diversity in the GA population. Experimental results show that the proposed algorithm can return multiple good quality solutions that also have minimal overlap with each other. Numerous solutions provide significant benefits when the test data contains none or missing values. Experiments were conducted using two publicly available time-series datasets. The feature sets are also applied to perform forecasting using a simple Long Short-Term Memory (LSTM) model, and the solution quality of the forecasting using different feature sets is analyzed. The proposed algorithm was compared with a popular optimization tool 'Basic Open-source Nonlinear Mixed INteger programming' (BONMIN), and a recent feature selection algorithm 'Conditional Mutual Information Considering Feature Interaction' (CMFSI). The experiments show that the multiple solutions found by the proposed method have good quality and minimal overlap.

INDEX TERMS Feature selection, genetic algorithm, machine learning, deep learning, optimization methods, forecasting.

I. INTRODUCTION

Time-series data contains observations recorded at regular time intervals. The record may contain one (univariate) or multiple variables (multivariate). The term 'lag-variables' is used to denote those variables that hold previous data. An example of univariate data is the record of water demand for every 15 minutes at a water-supply plant [1]. A stationary time-series refers to the time-series whose statistical properties (mean, variance and autocorrelation) remain unchanged over time. We can apply different types of transformations to convert a non-stationary time-series into a stationary one. Most of the forecasting methods uses stationary time-series and produce reliable results. Reliable forecasting using time-series data has a significant commercial benefit in many organizations. For example, a power distribution company may want to predict the power demand for the next few minutes, months, or even years, to adjust its generation capability. Although simple methods such as linear regression can do forecasting, they are usually not as reliable. The reliable methods of forecasting include: Auto-Regressive Moving Average (ARMA), Neural Network (NN), and different types of Deep Learning (DL)-models. The introduction of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) cells have significantly improved the performance of DL models to learn the complicated time-data relationships and make reliable predictions. Recent research shows that in forecasting, DL-based models have outperformed the NN and ARMA based models [1], [2]. Some examples that employed NN or DL based methods and univariate time-series data for

The associate editor coordinating the review of this manuscript and approving it for publication was Shagufta Henna.

reliable forecasting are: Forecasting the power generation of a wind-farm for the next 24 hours [3], short-term forecasting (i.e., up to nine days) of the residential electrical power load using data of the previous three months [4], and, short term (15 min and 24 hours) forecasting of water demand using data collected from district metering areas [1], to name a few.

In Machine Learning (ML) and DL, the lag-variables are termed as features. The feature selection problem can be thought of as a pre-processing step in the application of any ML or DL based forecasting method, and consists of the selecting a subset of features that can capture the characterises of all lag-variables [5]. Features selection has many benefits, such as: (i) it reduces the training time of the model; (ii) it helps in simplifying the complexity of forecasters; (iii) it improves the accuracy of the model; and (iv) it also avoids over-fitting by eliminating unnecessary variables from the feature set. For instance, when some lag-variables contain noise, then their inclusion in the feature set causes over-fitting of the model. The feature selection can eliminate variables containing noise from the feature set. In the absence of feature selection, the forecaster might need to deal with a large number of lag-variables for good quality forecasting that could lead to the well-known curse of dimensionality problem [6].

Feature selection techniques can be categorized into three types: (i) Filter methods; (ii) Wrapper methods; and (iii) Embedded methods. Filter methods assume complete independence between the data and the learning model and employ a statistical metric that has no connection with the learning model to rank and select the best features [7]. Wrapper based methods consider the problem as a *search problem* and evaluate different combinations of a subset of features until the best one is found. This method is time/computationally intensive because it involves training of the DL or ML model for each new feature combination. Finally, the embedded methods consider the feature selection as part of the learning process [8]. The filter methods are the fastest among all types and have shown good performance in many applications [7].

Filter based methods often employ techniques from information theory such as Mutual Information (MI), Joint Mutual Information (JMI), and Conditional Mutual Information (CMI). These techniques are instrumental in selecting a subset of features that improve the quality of the DL model. In information theory, the MI of two random variables is a measure of the mutual dependence between them. Although Pearson-correlation can also serve the same purpose, it can only determine linear relationships. Therefore, it is rarely employed in feature selection. In the context of time series data, one variable is the value at time t, and the other variable is any value at time t-x, where x is a non-zero positive integer. The JMI and CMI techniques involve three or more variables. JMI refers to the dependence of the output on composite variables, for example, the dependence of the values at time t on the composite of values at $t - x$ and $t - y$, where x and y are different, non-zero positive integers. The CMI metric computes the dependence of the output variable on a second variable when a third variable is also known, for example, the dependence of the values at time t on the values at $t - x$, when the values at time $t - y$ are already known.

The number of features is usually large, and therefore, the problem becomes intractable. Both constructive and iterative heuristics are useful in solving the feature selection problem [3], [9]–[11]. In practice, data often contains noise or missing values. The application of population-based metaheuristics have an advantage that they provide robustness again noise and missing data by determining alternate feature selection sets of almost equal quality. The population-based metaheuristics return a population (or archive) of unique solutions. By enforcing diversity in population, they can ensure that all solutions are unique and also have good values of the feature selection metric.

The contribution of this work is the obtaining of multiple good quality unique feature selection sets. To accomplish this we employ GA. Feature selection metrics such as MI, JMI, and Minimal Redundancy and Maximal Relevance (mRMR) are used as objective functions. During the selection of the next generation in GA, we include a bias along with the fitness function using the overlap coefficient to enforce diversity in the population. We employed multiple metrics (MI, JMI, and mRMR) because all of these can contribute to reliable forecasting, and we demonstrate that our proposed heuristic remains useful with all of these metrics.

This article is organized as follows. In the next section (Section II) we present some recent heuristics that have been employed which use CMI or MI to perform feature selection. In Section III, we present some relevant definitions. Section IV presents the proposed heuristic in detail. Complexity analysis is presented in Section V. In Section VI, we show the experimental results and their analysis. The last section contains the conclusion.

II. RELATED WORK

The feature selection is an NP-hard problem [12], [13] and therefore it should be solved using heuristic or metaheuristic algorithms. To find better heuristics for the feature selection problem, Battiti et al. introduced the use of MI metrics [14]. Since then, many new methods based on MI and CMI have been developed. In this section, we summarize some recent work in the area of feature selection methods based on MI and CMI metrics. We include both forecasting and classification methods because a majority of the previous research focused on classifiers. The forecasting problem is different from the classification problem in the following aspects. In the forecasting problem, the data items spans over a significant amount of time, and hence, the number of possible features can be huge. In the classifier, the number of attributes (i.e., the maximum features) is not huge. The outcome of the forecaster has a vast range, whereas, in classifiers, the number of classes is a small number, such as from 3 to 10 [11].

Naghibi et al. modeled the NP-hard feature selection problem as a (0-1)-quadratic integer programming and also relaxed that to a semi-definite programming problem which is a convex optimization problem [9]. The search consists of forward and backward selection heuristics. The forward search is a constructive method which iteratively adds new features in the selection set, and the backward search iteratively removes redundant features from the selection set.

Sensitivity analysis [15] has also been used in feature selection. Barraza et al. compared CMI and the sensitivity analysis approach for the feature selection problem [10]. In sensitivity analysis, we determine the significance of each feature by varying its value within a permissible range and observe the effect on the output (i.e., forecast). Barraza et al. compared a simple constructive method of feature selection based on MI with a method based on sensitivity analysis, and reported that the results of these approaches are quite different from each other. They found that both approaches have advantages and disadvantages. The sensitivity analysis approach is more suitable in the case of a lesser number of available features [10].

Babel et al. applied the MI based feature selection in rain forecasting [16]. They compared the MI based approach with two conventional methods. The first approach manually selects features from temporal data of the past rainfalls, and the second approach uses the meteorological data, other than the rainfall data, in feature selection. In their experiments, the MI-based method performed better than the conventional approaches. The MI metric helps in selecting relevant features and has been successfully applied in many applications such as feature selection in the problem of recognition of human gait [17], which has a large number of features.

In CMI-based methods, the objective function contains terms to account for the relevancy and redundancy of features. Liang et al. pointed out that that interaction among features is also common in classification data sets [11]. Therefore, the objective function should also include a term to account for the interaction of any feature with the selection set. They defined interaction as follows: A new feature can have interaction with the selection set when the value of its CMI with the output, given a selection set, is higher than the value of its MI with the output. They named their algorithm as Conditional Mutual Information Considering Feature Interaction (CMFSI), and evaluated its performance on several classification problems.

III. PROBLEM DEFINITION

A. PRELIMINARIES

Let X denote a random variable that can have up-to m discrete values,i.e., $X = \{x_1, x_2, \ldots, x_m\}$. The entropy of X measures its uncertainty. Let us consider a simple example to understand the concept of entropy. Suppose we have time-series data of temperatures of two cities A and B. The weather of city A remains almost constant and hot throughout the year. On the other hand, city B has all seasons and irregular rains, storms, and cold waves. The time-series data of city A has fewer number of distinct values as compared to the data of city B. Therefore, the time-series data of city B has a high entropy as compared to city A. Mathematically, it can be given by:

$$H(X) = -\sum_{i=1}^{m} p(x_i)\log p(x_i) \tag{1}$$

$p(x_i)$ represents the probability of x_i. The joint entropy of two random variables X and Y (where $Y = \{y_1, y_2, \ldots, y_n\}$) is given by:

$$H(X, Y) = -\sum_{i=1}^{m}\sum_{j=1}^{n} p(x_i, y_j)\log p(x_i, y_j) \tag{2}$$

$p(x_i, y_j)$ refers to the joint distribution probabilities of x_i and y_j. The conditional entropy of X when the output of Y is known, can be given by

$$H(X|Y) = -\sum_{i=1}^{m}\sum_{j=1}^{n} p(x_i, y_j)\log p(x_i|y_j) \tag{3}$$

$p(x_i|y_j)$ indicates the conditional probability of x_i when the outcome of y_j is known. MI detects relationship between two random variables (X, Y) and also measures it quantitatively. Let us consider a simple example to understand the concept of MI. Suppose that we have two variables X and Y. X represents the score of students in the final exam, and Y denotes the score of students in the class tests. From the scores of class tests, we can get information about the scores of students in the final exam, and vice versa. Therefore, X and Y have a high value of MI. Mathematically, it can be given by

$$I(X; Y) = \sum_{i=1}^{m}\sum_{j=1}^{n} p(x_i, y_j)\log\frac{p(x_i|y_j)}{p(x_i)} \tag{4}$$

Using (1) and (2), we can re-write the above equation as:

$$I(X; Y) = H(X) - H(X|Y) \tag{5}$$

The CMI involves at-least three random variables (X, Y, and Z), and is equal to the mutual information shared by X and Y when Z is known, i.e.,

$$I(X; Y|Z) = H(X|Z) - H(X|Y; Z) \tag{6}$$

We can extend MI to more than two variables to quantitatively express the interaction among different random variables [18]. For three variables (X, Y, Z), the multi-information can be given as

$$I(X; Y; Z) = \begin{cases} I(X; Z) + I(Y; Z) - I(X, Y; Z) \\ I(Z; X|Y) - I(Z; X) \\ I(Z; Y|X) - I(Y; Z) \\ I(X; Y|Z) - I(X; Y) \end{cases} \tag{7}$$

$I(X, Y; Z)$ denotes the mutual information of the joint distribution of X and Y relative to Z.

B. FEATURE SELECTION PROBLEM AND PERFORMANCE METRICS

Given a vector of past observations $D = \{d(t-1), \ldots, d(t-N)\}$ (where, t is the time variable) of any physical quantity such as water, or power demand, wind flow, etc., the forecasting problem consists of predicting the values of the physical quantity for the current time t, which is denoted by $d(t)$. Assuming that the selection set of features contains up to K features ($K \leq N$), the selection set can be represented as $F = \{t-\alpha_1, t-\alpha_2, \ldots, t-\alpha_K\}$, where α_i ($\forall\ i = 1$ to K) are non-zero positive integers and indicate different time steps. The feature selection methods should determine suitable values of the set $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$. The goal of feature selection method is to select a minimum number of features (i.e., smallest K value) that carry maximum information about the output. Many performance metrics exist for the feature selection problem that evaluate a selection set based on two criteria: (i) relevancy, and, (ii) redundancy, of the features. The relevancy of a feature is the dependence of the output on it. Redundancy refers to the situation when the feature set contains several features, and although each feature individually has a strong relationship with the output, the relationship of some features becomes unnecessary because of the other features present in the selection set.

Some popular information-theory based metrics used in feature selection methods are: (i) MI [19]; (ii) JMI [20], and (iii) mRMR [21]. Mathematically, these objective functions can be expressed as follows.

$$\mathrm{MI}(F) = \sum_{i=1}^{K} I(d(t-\alpha_i); d(t)) \tag{8}$$

$$\mathrm{JMI}(F) = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} I(d(t-\alpha_i), d(t-\alpha_j); d(t)) \tag{9}$$

$$\mathrm{mRMR}(F) = \sum_{i=1}^{K} I(d(t-\alpha_i); d(t)) - \frac{1}{K-1} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} I(d(t-\alpha_i); d(t-\alpha_j)) \tag{10}$$

Equation (8) shows a simple MI measure which is the summation of the MI value of each feature with the output (or $d(t)$). In (9), $I(d(t-\alpha_i), d(t-\alpha_j); d(t))$ denotes the JMI and it can re-written using (7) as follows: $I(d(t-\alpha_i), d(t-\alpha_j); d(t)) = I(d(t-\alpha_i); d(t)) + I(d(t-\alpha_j); d(t))$ - $I(d(t-\alpha_i); d(t-\alpha_j); d(t))$. The JMI measure is suitable to select the most relevant features. However, in some cases, it also mis-estimates the redundancy of features [22]. In the mRMR metric, as shown in (10), the relevance of the features is determined using the MI between the features and the output, and the redundancy of the features is determined using MI values between the features. We should find the selection sets that minimize the MI, JMI, and mRMR values. The computation of MI is a linear operation, whereas the computation of JMI and mRMR have a complexity of $O(N^2)$, using pre-computed values of the entropy (3) and MI (4).

IV. PROPOSED HEURISTIC

Genetic Algorithm (GA) is a popular population-based meta-heuristic for finding solutions to NP-hard combinatorial optimization problems. The population (of size M) comprises chromosomes, where each chromosome represents a complete *feature selection* set. The number of attributes in a chromosome is equal to the size of the feature selection set, i.e., K. We denote the population of GA using POP as follows.

$$\mathrm{POP} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \cdots & \alpha_{1,K} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \cdots & \alpha_{2,K} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \alpha_{M,1} & \alpha_{M,2} & \alpha_{M,3} & \cdots & \alpha_{M,K} \end{bmatrix} \tag{11}$$

In the above matrix, each row represents a chromosome (or a feature selection set), and variables $\alpha_{i,j}$ denote the feature $t_{\alpha i,j}$, where i, j are non-zero positive integers.

Algorithm 1 Overview of the GA Algorithm

```
Set parameters $M, p_\mu$;
Initialize the population POP with $M$ random solutions, and sets CHD, $\mathcal{F}^P$, $\mathcal{F}^C$ to { } (where $\mathcal{F}^P$ and $\mathcal{F}^C$ store the fitness values of chromosomes in the POP and CHD sets);
while Stopping criterion not reached do
    for i in 1 to M do
        Evaluate fitness of POP$_i$ and store it in $\mathcal{F}^P_i$
    $\eta = \{\}$, POP$' = \{\}$, $i = 1$ ;
    while $i \leq M$ do
        Apply the tournament selection method to select two chromosomes (POP$_x$ and POP$_y$) from POP, s.t. $x, y \notin \eta$ and are non-zero positive integers ;
        Apply the single-point crossover operator on POP$_x$, POP$_y$ to create two off-spring CHD$_i$ and CHD$_{i+1}$;
        Apply mutation operation with probability $p_\mu$ on CHD$_i$ and CHD$_{i+1}$;
        Insert $x$ and $y$ into $\eta$;
        i = i + 2 ;
    for i in 1 to M do
        Evaluate Fitness of CHD$_i$ and store it in $\mathcal{F}^C_i$
    Select $M$ chromosomes from POP ∪ CHD, and store them in POP
return POP;
```

Algorithm 1 gives an overview of GA. The algorithm has two parameters: M, and p_μ. M is the population size, and p_μ is the probability of mutation. GA returns more diverse solutions when the size of its population is large. However, a large population also increases the computation time. In our cases, the objective functions (MI, JMI, or mRMR) are computationally simple. Therefore, it is possible to use a large size population without any significant effect on the runtime of the heuristic. The variable CHD denotes the offspring chromosomes and has dimensions equal to that of POP.

The linear arrays F^P and F^C hold the fitness values of the chromosomes of the main population (POP) and offsprings (CHD). The population is initialized to random solutions, i.e., solutions in which features are chosen randomly from possible values (1 to N).

The stopping criterion of the main loop of the GA could be the maximum number of iterations or runtime. The evaluation step lies from lines 4 to 5 in the pseudo-code and computes the fitness of all chromosomes of the population. The fitness is the objective function, i.e., MI, JMI, or mRMR. The step after the computation of fitness values is to generate offsprings, which is an iterative process. The nested *while* loop that lies from line 7 to 12 performs the tasks of generating chromosomes. In each iteration of the nested *while* loop, the algorithm first selects two parent chromosomes from POP using binary tournament selection technique [23], and then applies the single-point crossover [23] to generate two new offspring referred to in the pseudo-code as CHD$_i$ and CHD$_{i+1}$. The offsprings also undergo mutation. The mutation operation tries to change the value of each attribute of the chromosome to a random value with a probability of p_μ. In the nested *while loop* , the parents chosen in any iteration do not take part in the binary tournament selections in the successive iterations. The number of offspring is kept equal to the population size M. The last step in the optimization loop is to select M chromosomes from the combined POP and CHD sets and replace the POP with the newly selected chromosomes. The selection of chromosomes is based on their fitness values. The method to select chromosomes consists of two steps as given below:

1) Elitism: Here the chromosome with best fitness value (i.e., the value of MI, JMI or mRMR metric) are always selected to remain in the population [24].
2) Biasing the fitness value: Here a bias factor that accounts for the diversity of the solution is added into its fitness values. The selection of chromosomes for the population of the next iteration uses modified fitness values.

In the following, we also discuss the algorithm that selects chromosomes for the population of the next iteration to keep the population size unchanged. Algorithm 2 describes the method using pseudo-code. In the first step, we combine the fitness values of the population and offspring into F^O, and initialize two sets SS and POP' to empty sets. The set SS stores the indices of chromosomes (from both POP and CHD sets) that constitute the population of the next iteration. The second and third lines of the algorithm perform elitism and make sure that the chromosome having the best fitness remains in the next iteration. In the fifth line, we modify the fitness values of the chromosomes based on a diversity-metric. The diversity metric used in this work is the 'overlapping-coefficient' [25]. The overlapping-coefficient between any two chromosomes X and Y can be defined as follows:

$$overlap(X, Y) = \frac{|X \cup Y|}{min(|X|, |Y|)} \tag{12}$$

In (12), the overlap coefficient between X and Y is equal to the ratio between the number of common features (i.e., attributes having identical values) in them to the size of the chromosome which is equal to K in this work. The average overlap coefficient is given below

$$\widehat{overlap}(X) = \frac{\sum_{Y \in \{POP \cap CHD - X\}} overlap(X, Y)}{M} \tag{13}$$

In (13), the average overlapping-coefficient of X is equal to the sum of the overlapping-coefficient of X with all chromosomes present in the POP and CHD sets, excluding, the overlapping-coefficient of X with itself,

In Algorithm 2, lines 5 to 7 indicate that we choose up-to M-1 solutions from the combined POP and CHD sets that have high fitness values for the population of the next iteration.

Algorithm 2 Procedure to Select Chromosomes for the Population of the Next Iteration

```
1 $F^O = F^P \cup F^C$, and SS = {} POP' = {} ;
2 Find the maximum fitness value and its index $i$ in $F_i^O$;
3 Insert $i$ into $SS$, and change the value at index $i$ in $F_i^O$ to
  $-\infty$ ;
4 Modify all fitness values in $F^O$ by adding the average
  overlapping-coefficient into them (13) ;
5 Find the best $M - 1$ values and their indices in $F_i^O$;
6 Insert the indices corresponding to the best $M$-1 fitness
  values into $SS$;
7 Fill the set POP' with the chromosomes of POP and
  CHD sets that have their index present in $SS$;
8 set POP = POP';
9 return POP;
```

V. TIME COMPLEXITY ANALYSIS

In this section, we present a brief analysis of the time complexity of the proposed heuristic. The proposed heuristic is described using two algorithms, Algorithm 1 and Algorithm 2. In Algorithm 1, the time complexity of the evaluation of fitness values of the population or offspring is as follows: When the objective function is MI, then it is $O(MK)$, and when the objective function is equal to JMI or mRMR then it is $O(MK^2)$. A call of tournament selection includes a linear operation of complexity $O(M)$ to split POP into two parts. The determination of the best fitness values in each part can be implemented using max-heap that has a complexity of $O(M)$. Therefore, the complexity of binary tournament selection is $O(M)$. The application of the single-point crossover operator includes the task to determine a random point in the chromosomes, which is a constant time operation. The crossover operation is a linear operation of complexity

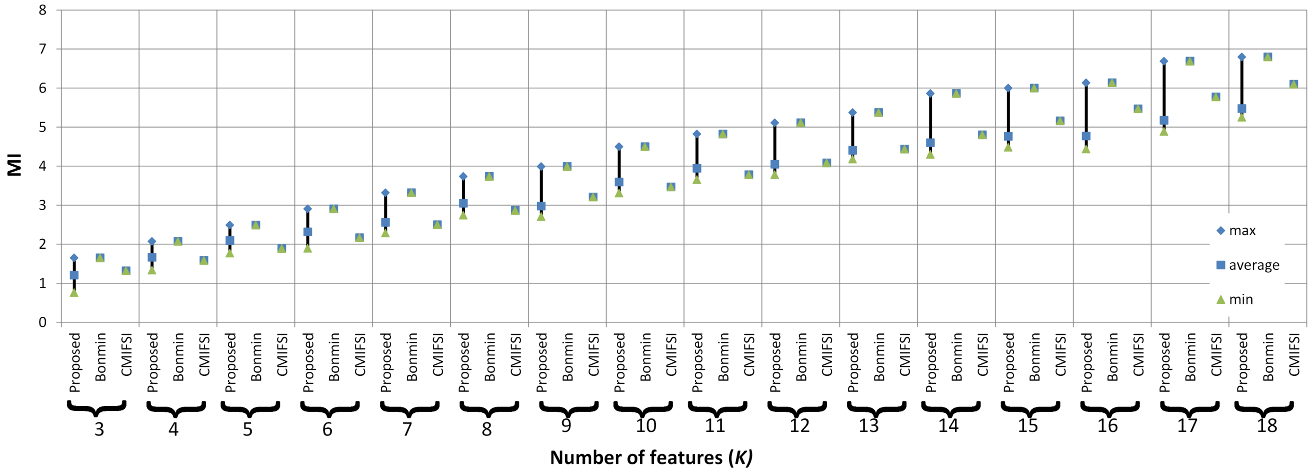


FIGURE 1. Results of the optimal feature selection using the MI metric on the test problem daily minimum temperature.

$O(M)$. The mutation is also a linear operation and has a time complexity equal to $O(M)$. The overall time complexity of the generation of up to M offspring is equal to $O(MK)$, because the genetic operations are linear in nature.

Algorithm 2 contains many memory copy steps that have constant time complexity. The task to determine the maximum value in F^O can be implemented using a max-heap, which has a time complexity equal to $O(M')$, where $M' = 2M$. The step to modify the fitness values of all chromosomes is a linear operation of complexity $O(M')$. The task to find $M - 1$ best values from F^O can be implemented using the heap-sorting which has a time complexity of $O(M' \log_2 M')$. Therefore, the time complexity of Algorithm 2 is $O(M' \log_2 M')$. The time complexity analysis of the GA shows that the evaluation of the fitness values is computationally time-intensive than any other part of the algorithm. The term 'evaluation count' is used to represent the number of times the objective function is computed before the algorithm converges to its best solution.

VI. EXPERIMENTAL RESULTS

This section shows and analyzes the experimental results of our design of the GA for solving the feature selection problem. We compare the performance of the proposed work with two existing heuristics: (i) Basic Open-source Nonlinear Mixed INteger programming (BONMIN) [26], [27], and, (ii) CMIFSI [11]. BONMIN is a popular open-source tool for solving Mixed Integer Non-Linear Programming problems, and is a part of several widely used optimization integrated development environments (IDEs) such as AMPL IDE [27]. The CMFSI is a feature selection method based on conditional mutual information proposed recently by Liang *et al.* [11]. Two time-series data sets from a public repository, Machine Learning Mastery [28] were used. The first dataset is the daily record of minimum temperature from 1981 to 1990 for the city of Melbourne, Australia. The second dataset contains the history of the monthly count of the number of sunspots from 1749-1983 of Zurich, Switzerland. In this article, we refer to the first dataset as "minimum-temperature", and the second dataset as "monthly-sunspots."

TABLE 1. Parameters of the LSTM model.

parameter	value
# of inputs	K (i.e., number of features)
Hidden layer	10 LSTM neurons
# of output	1
# of epochs	10
Training set	70%
Test set	30%

In the experiments, we employed the feature selection sets obtained by the proposed and existing methods to perform forecasting using the datasets mentioned above. Table 1 lists the parameters of the LSTM model used in this work. The LSTM model consists of three layers. The input layer has units equal to the feature size (K), the hidden layer consists of up to ten LSTM neurons, and the model has one output that contains forecast for the time t. The datasets are divided into training and test sets in proportions specified in Table 1. The quality of the forecaster is equal to the Root Mean Squared Error (RMSE) between the forecasted values and the actual values. A small value of RMSE indicates good quality of selected features. The quality of feature selection set obtained from LSTM forecaster is expressed as LSTMQ.

A. PERFORMANCE ANALYSIS USING FEATURE SELECTION METRICS

In the conducted experiments, we vary the size of the feature set (K) from 3 to 18, keeping the total number of features (N) fixed to 100. The parameter values of the GA algorithm are as follows: $M = 150$, and $p_\mu = 0.05$.

Figs. 1, 2, and 3 show the results of the feature selection metrics (MI, JMI, and mRMR) for the proposed method,

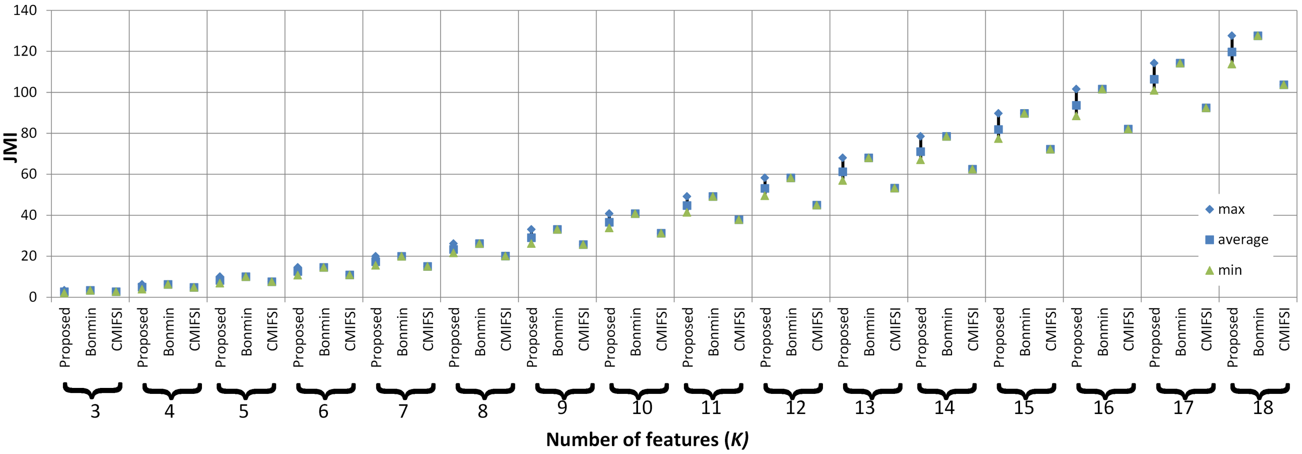


FIGURE 2. Results of the optimal feature selection using the JMI metric on the test problem daily minimum temperature.

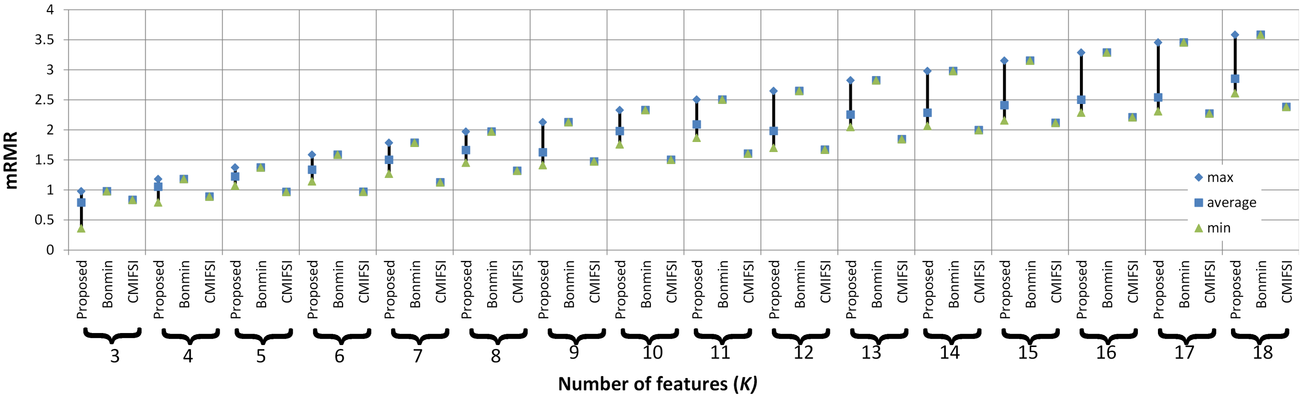


FIGURE 3. Results of the optimal feature selection using the mRMR metric on the test problem daily minimum temperature.

the BONMIN tool, and CMFSI algorithm. The x-axis indicates the number of features (K), and the y-axis shows the values of the feature selection metrics. Each result has three attributes: max, average, and min. The max, average, and min indicate the maximum value, average value, and minimum value of the set of solutions return by the proposed algorithm. Although the proposed algorithm has a population size of 200, we used the best 150 solutions (or solutions of the first three quartiles) in terms of the feature selection metrics for analysis. The BONMIN and CMFSI return only one solution, and, hence, the max average and min point to the same value. From the plots it can be observed that the best solution of the set of solutions returned by the proposed algorithm in terms of any feature selection metric is equal to the solution returned by the BONMIN tool and slightly better than the CMFSI algorithm. The plots also show that although the proposed algorithm returns a diverse set of solutions, there is a small difference in the values of their feature selection metrics. The results indicate that one can determine numerous alternative feature sets that are almost equal to each other in terms of different feature selection metrics.

TABLE 2. Average LSTMQ values for the feature sets found by BONMIN and CMMFSI.

	K	LSTMQ	
		BONMIN (Q_B)	CMFSI (Q_C)
Daily Minimum Temperature	3	(2.31,2.30,2.31)	2.31
	6	(2.30, 2.31,2.25)	2.33
	9	(2.31,2.30,2.24)	2.33
Monthly-sunspots	3	(18.37,18.43,20.10)	22.67
	6	(18.95, 19.25,19.96)	20.52
	9	(18.48, 19.02, 20.01)	22.80

B. PERFORMANCE ANALYSIS USING LSTMQ

In this sub-section, we first determine the forecasting quality (i.e., LSTMQ) of the feature sets returned by the BONMIN tool and CMFSI algorithm, and then analyze the forecasting quality of the proposed algorithm with the help of the forecasting quality of the BONMIN tool, and CMFSI algorithm. BONMIN returns a different solution for each combination of K and a feature selection metric, whereas CMFSI returns different solutions for different values of K. Since the training of DL models is NP-complete [29], we executed up to twenty trials of training-testing for each feature set found by BONMIN tool and CMFSI algorithm. Table 2 shows

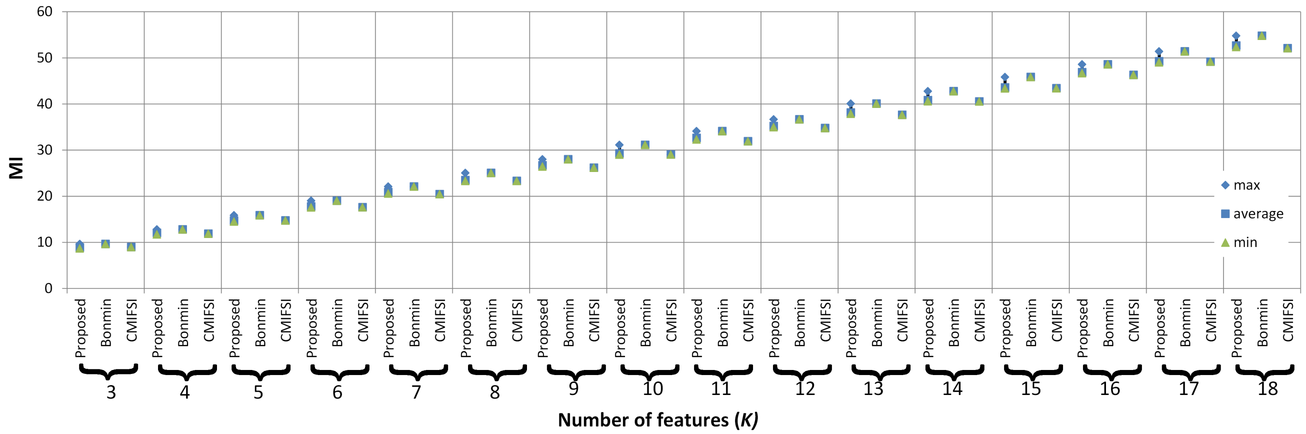


FIGURE 4. Results of the optimal feature selection using the MI metric on the test problem monthly sunspots.

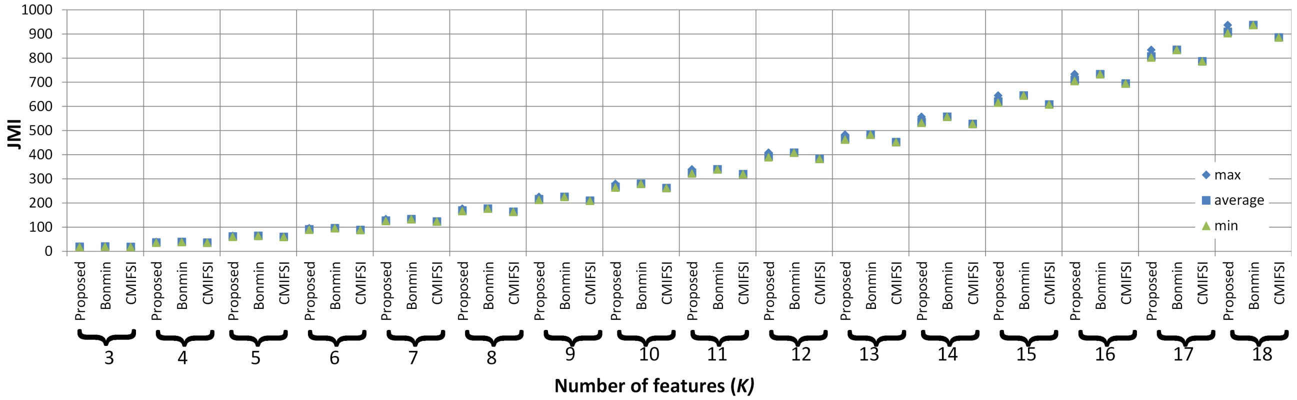


FIGURE 5. Results of the optimal feature selection using the JMI metric on the test problem monthly sunspots.

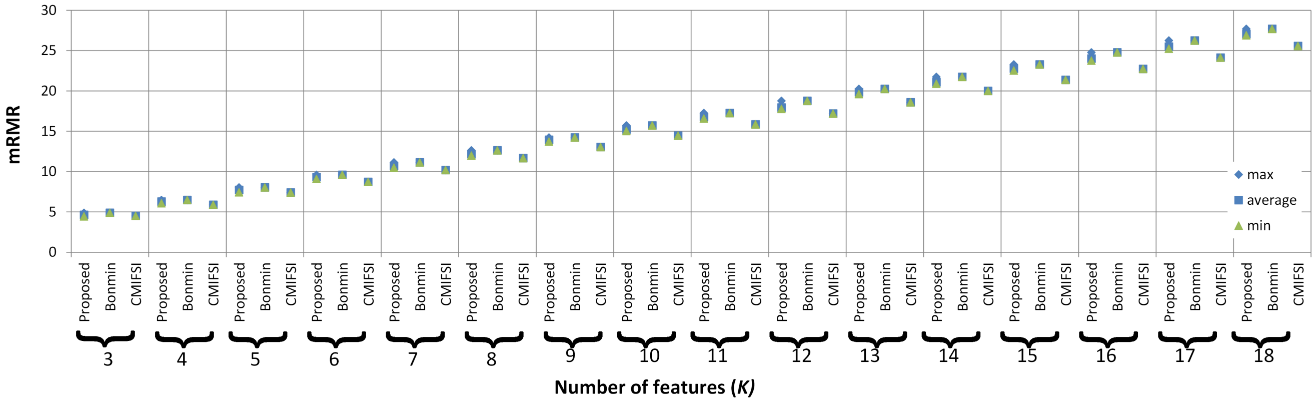


FIGURE 6. Results of the optimal feature selection using the mRMR metric on the test problem monthly sunspots.

the average LSTMQ values of all trials for the feature sets returned by the BONMIN tool and CMFSI algorithm. Q_B denotes the average LSTQM values of BONMIN, and Q_C indicates that of CSFSI. The column Q_B contains three values (a, b, c), where a, b, and c denote the average LSTQM values when the feature selection metrics are: MI, JMI, and mRMR. In this subsection, the population size of GA (M) is equal to 100, and the analysis uses all solutions of the population. The sizes of the feature sets (K) are 3, 6, and 9.

The remaining part of this subsection discusses the analysis of the forecasting quality of the different solutions returned by the proposed algorithm. We trained and tested the LSTM

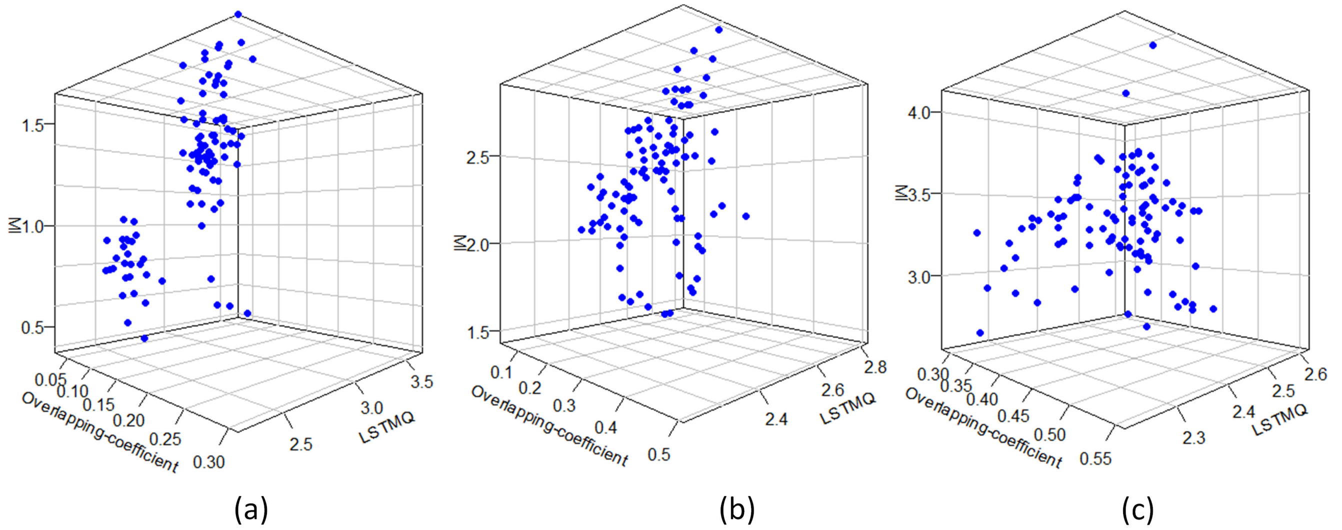


FIGURE 7. Relationship between the MI values, LSTMQ, and overlapping-coefficient when the test problem is daily-minimum temperature, and the number of features is equal to: (a) $K = 3$, (b) $K = 6$, and (c) $K = 9$.

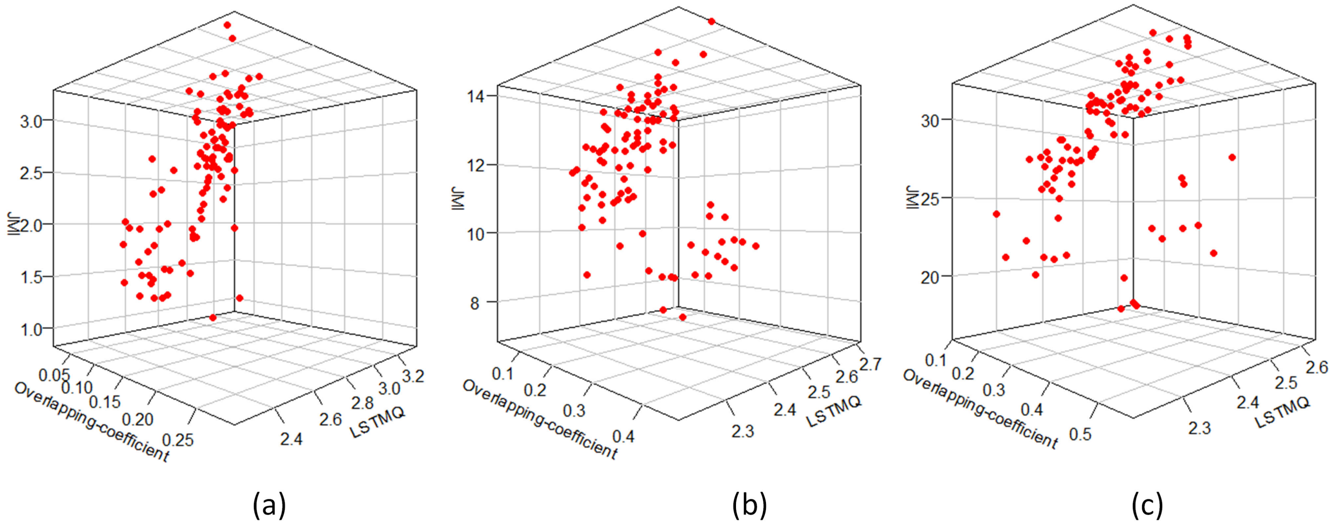


FIGURE 8. Relationship between the JMI values, LSTMQ, and overlapping-coefficient when the test problem is daily-minimum temperature, and the number of features is equal to: (a) $K = 3$, (b) $K = 6$, and (c) $K = 9$.

model using each solution (or chromosome) generated by the proposed heuristic. In the following paragraphs we present a brief analysis of their characteristics.

Figs 7(a)-(c), show the results for the test problem 'daily minimum temperatures', when the feature selection metric is MI, and the value of K varies from 3 to 9. Fig. 7(a) shows that for $K = 3$ up-to 25% solutions have LSTMQ values better than Q_B and Q_C (where, Q_B, Q_C equal to 2.30). For these 25% solutions, MI values lie in the range 1.2 and 1.65, and overlapping-coefficient values lie in the range 0.24 and 0.31. Please refer to Table 2 for the value of Q_B, and Q_C. In Fig. 7(b), for the value of K equal to 6, we notice that an increase in feature size improves the forecasting quality of the proposed algorithm. Now, up to 66% solutions have quality better than or equal to Q_B, and 74% solutions have quality better than or equal to Q_C (where, $Q_B = 2.3$ and $Q_C = 2.32$). The overlapping-coefficient and MI values of these solutions lie in the following ranges: 0.24 to 0.51, and 1.9 to 2.91, respectively. In Fig. 7(c), the percentage of solutions having LSTMQ value better than Q_B, and Q_C increases to 77% and 80% respectively, and the values of the overlapping-coefficient of these solutions lie in the range of 0.28 and 0.56.

Figs. 8(a), (b), and (c) show the results when the JMI metric used, and the test problem is 'daily minimum temperature'. From Figs. 8(a), (b), and (c), we can find out

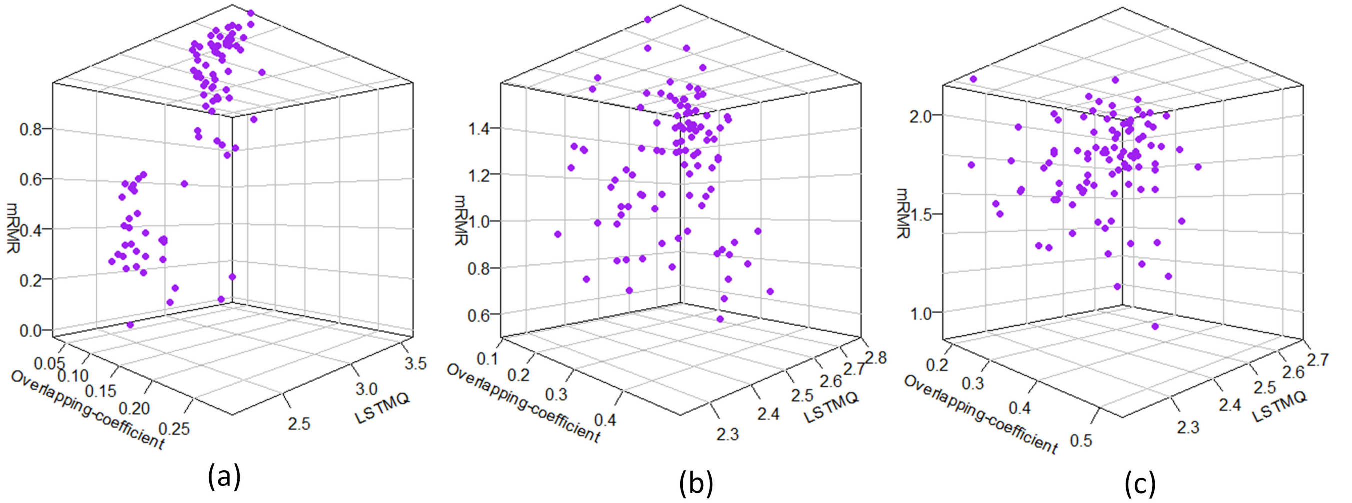


FIGURE 9. Relationship between the mRMR values, LSTMQ, and overlapping-coefficient when the test problem is daily-minimum temperature, and the number of features is equal to: (a) $K = 3$, (b) $K = 6$, and (c) $K = 9$.

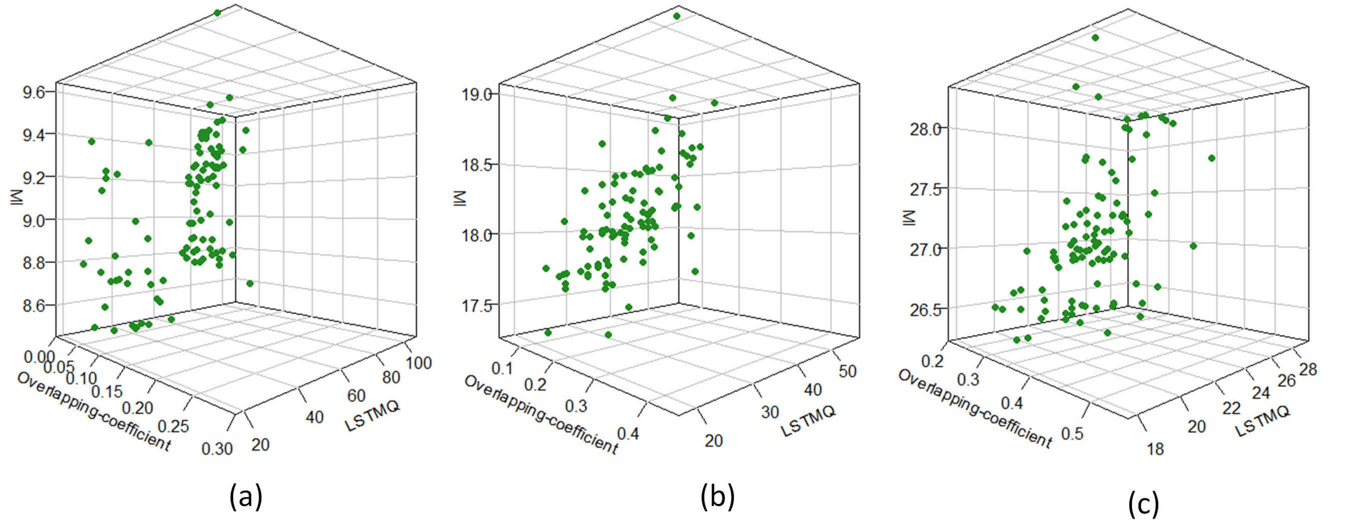


FIGURE 10. Relationship between the MI values, LSTMQ, and overlapping-coefficient when the test problem is monthly-sunspots, and the number of features is equal to: (a) $K = 3$, (b) $K = 6$, and (c) $K = 9$.

that the percentage of solutions having forecasting quality better than or equal to Q_B and Q_C, the percentages are as follows: (i) For $K = 3$, 6, and 9, up to 50%, 77%, and 68% solutions have LSTMQ values better or equal to Q_B; and (ii) For $K = 3$, 6, and 9, up to 50%, 78%, and 88% solutions have forecasting quality better than or equal to Q_C. The average overlapping-coefficient values of the solutions whose quality is better than Q_B and Q_C lie in the following ranges: (i) 0.23 to 0.29, for $K = 3$; (ii) 0.2 to 0.46, for $K = 6$; and (iii) 0.14 to 0.56,for $K = 9$.

Figs. 9(a), (b), and (c) contain the results when mRMR metric is employed and the test problem is 'daily minimum temperature'. Figs. 9(a), (b), and (c) show that the percentage of solutions having forecasting quality better than or equal to Q_B and Q_C are as follows: (i) For $K = 3$, 31% solutions are better than Q_B and Q_C (ii) For $K = 6$, 4% and 62% are better than Q_B; and Q_C; and (iii) For $K = 9$, 4% and 84% are better than Q_B; and Q_C, respectively. The average overlapping-coefficient values of the solutions whose quality is better than Q_B or Q_C lie in the following ranges: (i) 0.23 to 0.28, for $K = 3$; (ii) 0.32 and 0.49, for $K = 6$; and (iii) 0.23 and 0.0.53, for $K = 9$.

Figs 10, 11, and 12 show the relationship between a feature selection metric, LSTMQ, and overlapping-coefficient values for the test problem 'monthly-sunspots'. Figs 10(a), (b) and (c) show results when MI was used as an objective function, and it conveys the following information: (i) When $K = 3$,6 and 9, then 10%, 15%, and 23% solutions are better than or equal to Q_B, and 60%, 41%, and 90% solutions are better than Q_C; (ii) The average

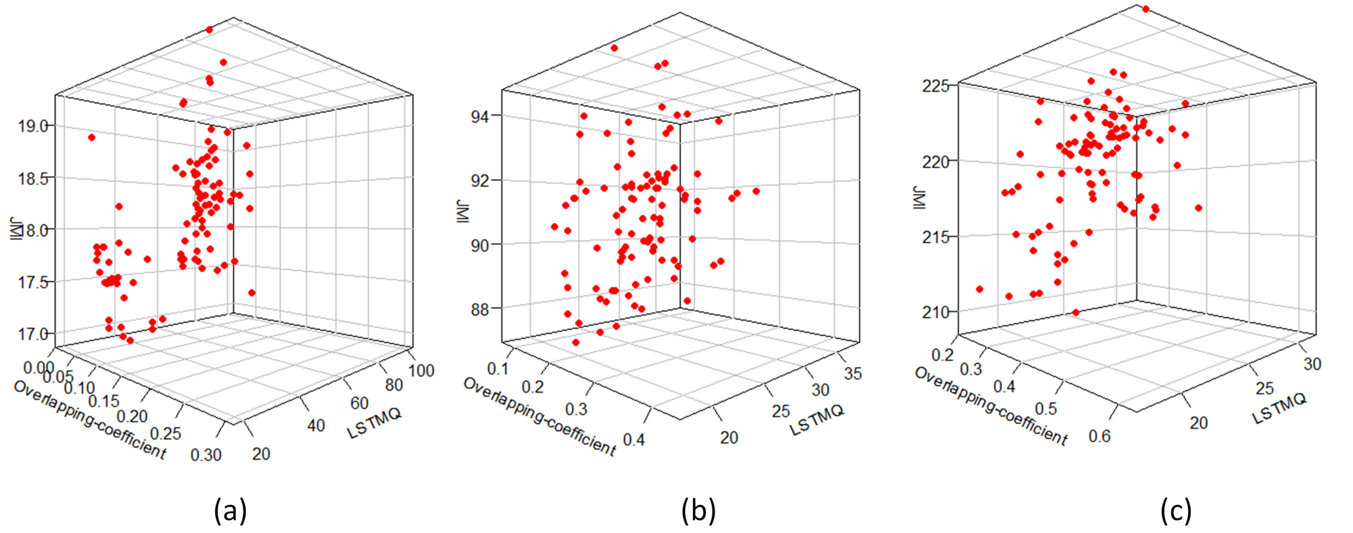


FIGURE 11. Relationship between the JMI values, LSTMQ, and overlapping-coefficient when the test problem is monthly-sunspots, and the number of features is equal to: (a) $K = 3$, (b) $K = 6$, and (c) $K = 9$.

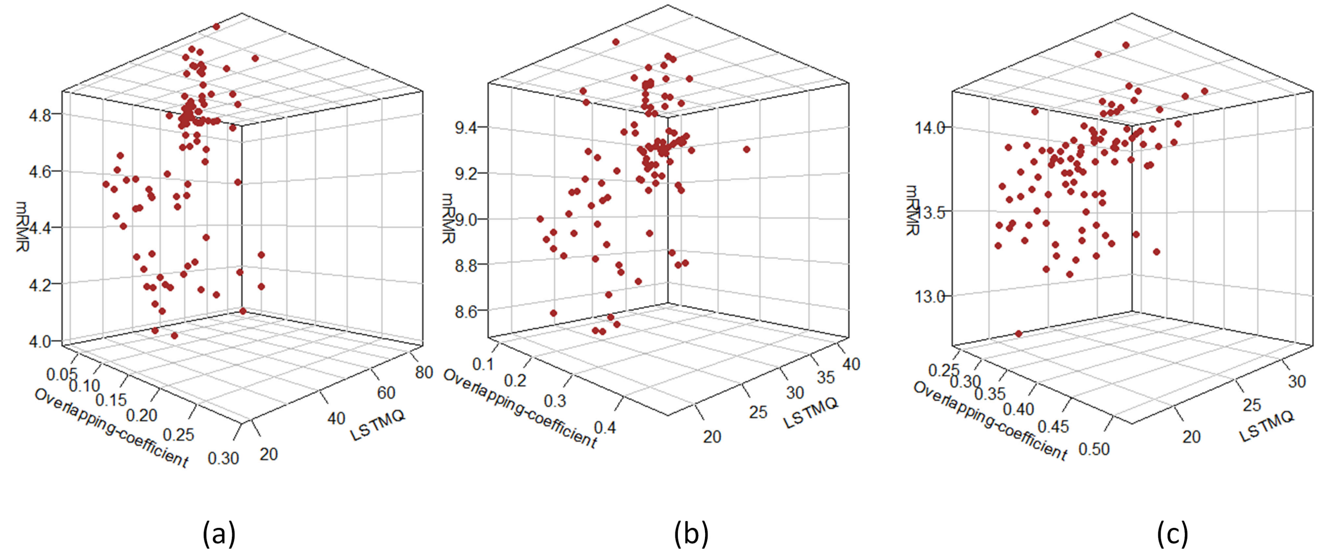


FIGURE 12. Relationship between the mRMR values, LSTMQ, and overlapping-coefficient when the test problem is monthly-sunspots, and the number of features is equal to: (a) $K = 3$, (b) $K = 6$, and (c) $K = 9$.

overlapping-coefficient values of the solutions which are better than both Q_B and Q_C lie in the following ranges: (a) 0.24 to 0.29, for K = 3; (b) 0.16 to 0.43, for K = 6; and (c) 0.29 to 0.55, for $K = 9$.

The results in Fig. 11 were obtained when the JMI metric was employed. The figure delivers the following important information: (a) When K = 3, then up 8% and 63% solutions are better than Q_B and Q_C, respectively. Furthermore, the average overlapping-coefficient of those solution lie from 0.24 to 0.28; (b) When K = 6, then up 27% and 43% solutions are better than Q_B and Q_C, respectively. The average overlapping-coefficient values of those solution lie from 0.17 to 0.44; and (c) When K = 9, then up 36% and 89% solutions are better than Q_B and Q_C, respectively. The average overlapping-coefficient of those solution lie from 0.22 to 0.63.

Finally, Figs 12(a),(b) and (c) show the results when the mRMR metric is used and the test problem in monthly-sunspots. The plots show that when $K = 3$, 6, and 9, then 31%, 60%, and 61% solutions of the population have LSTMQ values better than both Q_B and Q_C. The average overlapping-coefficient values of the solutions that have LSTMQ values better than Q_B and Q_C lie in the following ranges: (i) from 0.21 to 0.28, for $K = 3$; (ii) from

0.17 to 0.47, for $K = 6$; and (iii) from 0.29 to 0.52, for $K = 9$.

From the results presented in this subsection, we can infer the following about the performance of the proposed algorithm. The proposed algorithm can make efficient use of the increase in the feature set size and can find out many good quality solutions. The uniqueness of solutions reduces with an increase in the feature size. However, there always exist some unique solutions which have minimal overlap with others. Figs. 13 and 14 show the average overlapping -coefficient values of the solutions when the value of the number of features (K) is large and equal to 9. The results indicate the presence of one or several solutions that have minimal average overlapping-coefficient value. Therefore, the solutions of minimal overlap with other solutions exist for both small and large values of K.

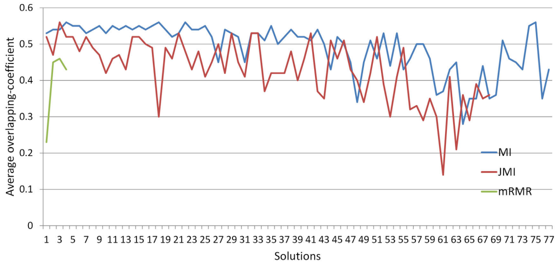


FIGURE 13. **The average overlapping-coefficient values of the solutions returned by the proposed algorithm with LSTMQ values better than Q_B and Q_C, when K = 9, and the test problem is 'Daily minimum temperature.'**

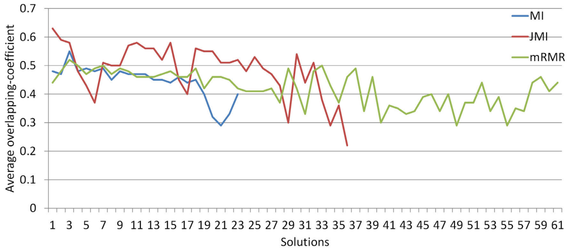


FIGURE 14. **The average overlapping-coefficient values of the solutions returned by the proposed algorithm with LSTMQ values better than Q_B and Q_C, when K = 9, and the test problem is 'Monthly-sunspots.'**

VII. CONCLUSION

Three prominent 'filter type' information theory-based metrics employed in the feature selection methods are as follows: MI, JMI, and mRMR. The problem of feature selection is NP-hard. The heuristics have been employed to solve it in the past. This work investigates using GA, a population-based metaheuristic, to find many alternative solutions of good quality. The alternate solutions provide possibility of selecting the best in case of noise or missing values in the test set. We applied GA and enhanced the fitness function by the addition of the overlapping-coefficient term. The GA implementation applies elitism in selecting chromosomes for the next generation. We conducted experiments using two publicly available time-series data sets and analyzed the results in detail. The experimental results on both problems showed that the proposed approach can find feature sets of better quality. The diversity of solutions is high in small size feature sets. However, when the feature size is large, then some solutions still have high diversity.

REFERENCES

[1] G. Guo, S. Liu, Y. Wu, J. Li, R. Zhou, and X. Zhu, "Short-term water demand forecast based on deep learning method," *J. Water Resour. Planning Manage.*, vol. 144, no. 12, Dec. 2018, Art. no. 04018076, doi: 10.1061/(asce)wr.1943-5452.0000992.

[2] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Acad. Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 324–328.

[3] S. Li, P. Wang, and L. Goel, "Wind power forecasting using neural network ensembles with feature selection," *IEEE Trans. Sustain. Energy*, vol. 6, no. 4, pp. 1447–1456, Oct. 2015.

[4] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.

[5] S. F. Crone and N. Kourentzes, "Feature selection for time series prediction—A combined filter and wrapper approach for neural networks," *Neurocomputing*, vol. 73, nos. 10–12, pp. 1923–1936, Jun. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231210000974

[6] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944968

[7] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, and M. Lang, "Benchmark for filter methods for feature selection in high-dimensional classification data," *Comput. Statist. Data Anal.*, vol. 143, Mar. 2020, Art. no. 106839. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016794731930194X

[8] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Adv. Bioinf.*, vol. 2015, 2015, Art. no. 198363.

[9] T. Fujie and M. Kojima, "Semidefinite programming relaxation for nonconvex quadratic programs," *J. Global Optim.*, vol. 10, no. 4, pp. 367–380, Jun. 1997. [Online]. Available: https://doi.org/10.1023/A:1008282830093

[10] N. Barraza, S. Moro, M. Ferreyra, and A. De La Peña, "Mutual information and sensitivity analysis for feature selection in customer targeting: A comparative study," *J. Inf. Sci.*, vol. 45, no. 1, pp. 53–67, Feb. 2019, doi: 10.1177/0165551518770967.

[11] J. Liang, L. Hou, Z. Luan, and W. Huang, "Feature selection with conditional mutual information considering feature interaction," *Symmetry*, vol. 11, no. 7, p. 858, Jul. 2019. [Online]. Available: https://www.mdpi.com/2073-8994/11/7/858

[12] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *J. King Saud Univ.-Comput. Inf. Sci.*, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1319157819304379

[13] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790613003066

[14] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, Jul. 1994.

[15] E. Borgonovo, *Sensitivity Analysis*, vol. 251. Cham, Switzerland: Springer, 2017.

[16] M. S. Babel, G. B. Badgujar, and V. R. Shinde, "Using the mutual information technique to select explanatory variables in artificial neural networks for rainfall forecasting," *Meteorol. Appl.*, vol. 22, no. 3, pp. 610–616, Jul. 2015, doi: 10.1002/met.1495.

[17] B. Guo and M. Nixon, "Gait feature subset selection by mutual information," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 1, pp. 36–46, Jan. 2009.

[18] W. J. Mcgill, ''Multivariate information transmission,'' *Psychometrika*, vol. 19, no. 2, pp. 97–116, Jun. 1954, doi: 10.1007/bf02289159.

[19] H. Peng and Y. Fan, ''Feature selection by optimizing a lower bound of conditional mutual information,'' *Inf. Sci.*, vols. 418–419, pp. 652–667, Dec. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025517308836

[20] P. E. Meyer and G. Bontempi, ''On the use of variable complementarity for feature selection in cancer classification,'' in *Applications of Evolutionary Computing*, F. Rothlauf, J. Branke, S. Cagnoni, E. Costa, C. Cotta, R. Drechsler, E. Lutton, P. Machado, J. H. Moore, J. Romero, G. D. Smith, G. Squillero, and H. Takagi, Eds. Berlin, Germany: Springer, 2006, pp. 91–102.

[21] H. Peng, F. Long, and C. Ding, ''Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[22] M. Bennasar, Y. Hicks, and R. Setchi, ''Feature selection using joint mutual information maximisation,'' *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8520–8532, Dec. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417415004674

[23] S. M. Sait and H. Youssef, *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems*, 1st ed. Los Alamitos, CA, USA: IEEE Computer Society Press, 1999.

[24] H. Du, Z. Wang, W. Zhan, and J. Guo, ''Elitism and distance strategy for selection of evolutionary algorithms,'' *IEEE Access*, vol. 6, pp. 44531–44541, 2018.

[25] H. F. Inman and E. L. Bradley, ''The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities,'' *Commun. Statist.-Theory Methods*, vol. 18, no. 10, pp. 3851–3874, Jan. 1989, doi: 10.1080/03610928908830127.

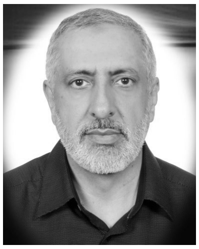
[26] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, ''An algorithmic framework for convex mixed integer nonlinear programs,'' *Discrete Optim.*, vol. 5, no. 2, pp. 186–204, May 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1572528607000448

[27] D. M. Gay, ''The AMPL modeling language: An aid to formulating and solving optimization problems,'' in *Numerical Analysis and Optimization*, M. Al-Baali, L. Grandinetti, and A. Purnama, Eds. Cham, Switzerland: Springer, 2015, pp. 95–116.

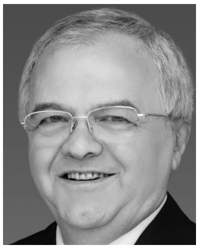
[28] Machine Learning Mastery. (2019). *Minimum Daily Temperatures Dataset*. Accessed: Nov. 11, 2019. [Online]. Available: https://machinelearningmastery.com/time-series-datasets-for-machine-learning/

[29] A. Blum and R. L. Rivest, ''Training a 3-node neural network is np-complete,'' in *Proc. 1st Annu. Workshop Comput. Learn. Theory (COLT)*, 1988, pp. 9–18. [Online]. Available: http://dl.acm.org/citation.cfm?id=93025.93033

UMAIR F. SIDDIQI (Member, IEEE) was born in Karachi, Pakistan, in 1979. He received the B.E. degree in electrical engineering from the NED University of Engineering and Technology, Karachi, in 2002, the M.Sc. degree in computer engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 2007, and the Dr.Eng. degree from Gunma University, Japan, in 2013. He is currently a Research Engineer with the Center of Communications and Information Technology Research, Research Institute, KFUPM. He has authored over 30 research articles in international journals and conferences. He also holds three U.S. patents. His research interests include algorithms, metaheuristic algorithms, soft computing, optimization, and deep learning.



SADIQ M. SAIT was born in Bengaluru. He received the bachelor's degree in electronics engineering from Bangalore University, in 1981, and the master's and Ph.D. degrees in electrical engineering from KFUPM, in 1983 and 1987, respectively. He is currently a Professor of computer engineering and the Director of the Center for Communications and IT Research, KFUPM. He has authored over 200 research articles, contributed chapters to technical books, and lectured in over 25 countries. He is also the principle author of two books. In 1981, he received the best Electronic Engineer award from the Indian Institute of Electrical Engineers, Bengaluru.



OKYAY KAYNAK (Fellow, IEEE) received the B.Sc. (Hons.) and Ph.D. degrees in electronic and electrical engineering from the University of Birmingham, Birmingham, U.K., in 1969 and 1972, respectively.

From 1972 to 1979, he held various positions within the industry. In 1979, he joined the Department of Electrical and Electronics Engineering, Bogaziçi University, Istanbul, Turkey, where he is currently a Professor Emeritus, holding the UNESCO Chair on Mechatronics and also a 1000 Talents Program Professor at the University of Science and Technology Beijing, China. He holds long-term (near to or more than a year) Visiting Professor/Scholar positions at various institutions in Japan, Germany, USA, Singapore, and China. He has authored three books and edited five and authored or coauthored almost 400 articles that have appeared in various journals, books, and conference proceedings. His current research interests include the fields of intelligent control and CPS.

Dr. Kaynak has been on many committees of the IEEE and was the President of the IEEE Industrial Electronics Society, from 2002 to 2003. He recently received the Chinese Government's Friendship Award and Humboldt Research Prize, both in 2016.