



A hybrid mobile call fraud detection model using optimized fuzzy C-means clustering and group method of data handling-based network

Sharmila Subudhi¹ · Suvasini Panigrahi¹

Received: 31 July 2016 / Accepted: 15 May 2018 / Published online: 26 May 2018
© The Author(s) 2018

Abstract

A novel two-stage fraud detection system in mobile telecom networks has been presented in this paper that identifies the malicious calls among the normal ones in two stages. Initially, a genetic algorithm-based optimized fuzzy c-means clustering is applied to the user's historical call records for constructing the calling profile. Thereafter, the identification of the fraudulent calls occurs in two stages. In the first stage, each incoming call is passed to the clustering module that identifies the call as genuine, malicious or suspicious. This is done by comparing the distance value of the new calling instance from the profile cluster centers against two predefined threshold values. The calls detected as genuine or malicious are not further processed. However, the call records that are found to be suspicious are additionally scrutinized in the second stage by a previously trained group method of data handling model for final decision making. The legitimate and forged labeled call records generated out of the clustering module are utilized for training the supervised classifier. Experimentation is done on a real-world call dataset to exhibit the effectiveness of the proposed model. A comparative analysis of the current approach with one of our earlier propositions and another recent fraud detection system clearly illustrates the efficacy of the developed model.

Keywords Fraud detection · Mobile phone call records · Genetic algorithm · Fuzzy C-means · Group method of data handling

1 Introduction

With the surge in the subscription of mobile phone services, the telecom service provider companies have been plagued with the problem of telecom fraud which occurs when a person employs deceitful techniques to successfully obtain the telephonic amenities freely or at a lower rate [18]. According to a survey conducted by the Communications Fraud Control Association (CFCA), the telecommunication industry has lost nearly \$46.3 billion worldwide in 2013 [38]. Another study done by the organization FFA UK (Financial Fraud Action United Kingdom) stated that due to various telephonic scams, the UK telecom companies have suffered a loss of £23.9 million in 2014, which is three times higher than that of the previous year [27]. The telecom fraud can be segmented into various types, out of which the superimposed fraud represents the most typical one that can be

defined as accessing a genuine subscriber's calling account to make malicious calls [10]. As per a report published by CFCA, the telecom companies worldwide have lost \$38.1 billion due to the fraudulent activities in 2015, out of which the superimposed fraud accounted for nearly 6% of the total amount [23]. Therefore, in this work, we aim at detecting this type of fraud since it constitutes a more bigger and riskier issue for the telecom business.

To handle such fraud cases, many researchers have developed various approaches by using different clustering and classification techniques [20,26,32,33]. The details of these methods have been discussed in the next section. It is found from the study of the literature that the existing methods have used the hard clustering techniques to build the subscriber's calling profiles. But such clustering methods are unable to capture the dynamic calling behavior of the user effectively due to its inability in managing the overlapped clusters. Furthermore, it is to be noted that a user may not follow a specific pattern while making a call. Therefore, the concept of fuzzy C-means (FCM) clustering has been deployed in this work so as to capture the uncertain behavior of subscribers. However, two prime issues faced by FCM is the random initialization

✉ Sharmila Subudhi
sharmilasubudhi1@gmail.com

¹ Department of Computer Science and Engineering, Veer Surendra Sai University of Technology, Burla, India

of the cluster centers and the tendency of its cost function to be stuck in a local optimum [7]. Hence, an evolutionary optimized algorithm, known as genetic algorithm (GA), is used on the fuzzy clusters to optimize their cluster centers for more accurate user profiling and, thereby, improving the performance of the fraud detection system (FDS).

Another major concern associated with most of the supervised classifiers used for telecom fraud detection is the estimation of various computational parameters needed for its proper functioning, which is a cumbersome and time-consuming procedure. Therefore, this paper emphasizes using the group method of data handling (GMDH) classifier for faster real-time fraud detection as it automatically determines the required input parameters [24]. The GMDH constructs a learning model on the relationship between the input and output variables of the dataset by considering few training parameters [5]. Furthermore, no user interference is required for establishing such relationship. This technique has successfully been deployed in different fields, such as attribute selection [1], financial prediction [34], pattern recognition and forecasting [45], and intrusion detection [1,3,36] as well.

Based on the observations as discussed, this paper introduces a novel anomaly-based hybrid FDS that can adapt to the dynamic calling behavior of the subscribers by self-learning its classifier parameters. Initially, the FCM clustering has been applied to the user's past call records for building their respective normal calling profiles. GA is then employed on the fuzzy clusters for generating optimized fuzzy clusters. A new calling instance is passed through the clustering module that classifies the transaction into either of the three different categories—genuine, fraudulent or suspicious—according to its distance value measured from the optimized cluster centers. If the call is detected as genuine or malicious, it is not processed further. However, if the call is found to be suspicious, then additional verification and final classification are made by applying a previously trained GMDH classifier.

The organization of the article proceeds as follows. Section 2 depicts the literature on superimposed mobile phone fraud detection. The background of various techniques implemented in our current work is presented in Sect. 3. The working methodology of the FDS has been described in Sect. 4. The results obtained from experimental analysis has been illustrated in Sect. 5. Finally, Sect. 6 concludes the paper summarizing the contributions and research outcomes.

2 Literature review

This section deals with the studies carried out with respect to superimposed telecom fraud detection.

The concept of latent Dirichlet allocation (LDA) probabilistic model for building normal user profile has been

used in an FDS developed in [32]. This paper has also used the Kullback–Leibler divergence (KL-divergence) technique between two LDA models for identification of illegitimate activities. Furthermore, the work suggested in [33] has employed a self organizing map (SOM) for demonstrating the significance of subscriber account visualization in the context of mobile phone fraud detection, while the illegitimate actions are finally identified by employing a threshold-based classification technique. Another FDS based on genetic programming (GP) has been developed in [20] for discriminating the illicit actions from the genuine ones. Additionally, four different attribute selection techniques have been used for choosing the important features from the historical call records of each user to construct five normal calling profiles. Finally, the discrimination of forged calling events is carried out by using the GP classifier.

The paper [26] presents an approach that identifies the fraudulent calls by initially forming groups of mobile phone users based on their calling instances present in the training set. A behavior pattern matching algorithm is then used for matching a new call record with the normal user groups. The call is marked as normal if maximum similarity is found; otherwise, it is labeled as malicious. The use of unsupervised quarter sphere support vector machine (QSSVM) has been suggested for identifying the fraudulent calls in [39]. The authors have modeled the user's normal calling profile by considering the spatiotemporal attributes along with other relevant features. The paper [21] demonstrates the usefulness of two clustering methods, namely, hierarchical agglomerative and K-means for identifying illicit actions in the calling profiles by constructing five subscriber profiles from their respective call records. Any sign of illegitimate activities found in the incoming call is analyzed by visualizing the clustering output generated from those profiles.

An approach proposed in [40] has used FCM and SVM on the past call records of each user for detecting fraudulent calls. The FCM clustering technique has been applied to certain calling features for user profile construction. The clustering outputs are then fed to SVM as input for building a trained SVM model, which then identifies a recent call record as a malicious one for not complying with the model. Another FDS developed for detection of forged calls in the call records has used the possibilistic fuzzy C-means (PFCM) clustering and hidden Markov model (HMM) in tandem [41]. PFCM has been initially applied to certain calling attributes for building the subscriber's normal calling profile. The parameter values required for training the HMM has been extracted from these profiles and a normal profile sequence has been produced. Similarly, another sequence has been generated from the trained HMM model for each new call and tested against the original profile sequence for final classification.

Based on the limitations identified in the existing work as discussed in Sect. 1, the current work proposes a hybrid mobile phone FDS that deploys GA-based FCM clustering for correct subscriber profiling and GMDH for effective fraud identification.

3 Background study

This section depicts the brief introduction of the techniques—GMDH, GA and FCM for understanding the working mechanism of the proposed system.

3.1 Genetic algorithm

The GA-based evolutionary optimization technique is first conceptualized in [22] by considering Darwin’s “Survival of the fittest” evolution theory. It is a natural genetic search algorithm which is iteratively used on an initial set of probable solutions, called as chromosomes, to produce the best pair of a solution. This is achieved by choosing a proper selection strategy, type of crossover and mutation operators [9]. Crossover takes more than one parent chromosome and produces a child, while mutation changes one or more than one gene. Thereby, a new group of solutions is identified from the old solution space while performing a global parallel search in each iteration. This procedure helps in the evolution of a population that are more acceptable to their domain than their previous individuals [9].

Two crossover methods—uniform crossover and n-point crossover are used to perform a crossover operation by combining any two selected individuals together to produce an offspring. A crossover rate parameter $p_c \in [0.6, 1.0]$ is used to represent the possibility of any two individuals to receive the crossover [4]. Three selection techniques—roulette-wheel selection, tournament selection and ranking selection have been used in GA for choosing a selection strategy required for performing crossover. Finally, mutation is applied to the chromosomes with a mutation rate $p_m > 1\%$ to instigate a little randomness so that the optimization procedure will not be stuck in the local optima.

3.2 Fuzzy C-means

Fuzzy C-Means focuses on finding suitable fuzzy groups for a dataset [7]. It takes the data instances as input and forms groups after assigning some membership values within the range of [0, 1] to them. The FCM algorithm can easily be adapted to the classes that are not well separated [25]. The objective function of FCM [7] can be expressed as:

$$\min J_m(U, V; D) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik}^m) B_{ik}(v_i, d_k), \quad (1)$$

owing to the conditions $\sum_{i=1}^c u_{ik} = 1 \forall k, 0 \leq u_{ik} \leq 1$. The cost function is denoted as J_m and $m > 1$ is a fuzzy weighting value. Usually, $m = 2$ is used for better clustering as the clusters tend to be crisp for $m = 1$ [7]. The membership matrix is $U = [u_{ik}]$, and $V = \{v_1, v_2, \dots, v_c\}$ is the vector of c cluster centroids, while the dataset $D = \{d_1, d_2, \dots, d_n\}$ contains n instances used for clustering. $B_{ik}(v_i, d_k)$ is any distance measure between an instance d_k and cluster center v_i . After giving the dataset with the required number of clusters (c) to FCM as input, it generates U (membership matrix) and V (cluster center matrix).

Although FCM has wide applicability in various domains [2,30,47], it suffers from the issue of random initialization of the cluster centers and the tendency of its cost function to be stuck in a local optima [7]. To overcome such limitations, several extensions of the traditional FCM such as intuitionistic fuzzy set [46], picture fuzzy set [42] and kernel fuzzy set [29,37] have been proposed. However, intuitionistic FCM takes more number of iterations to find out the number of cluster centers than FCM, resulting in high computational time [46]. Similarly, in case of picture fuzzy set, an extra exponent parameter value is required to be set to obtain best fuzzy cluster sets, thus requiring more computational time [42]. Likewise, for kernel-based FCM, the problem lies in selecting the best kernel to find out the optimal distance of each point from the cluster center, which is a quite tedious process [29,37]. Hence, we have chosen the classical FCM algorithm in the current work rather than its variants and applied GA on it for optimizing the cluster centers by searching a global optimum to make the clustering approach more robust.

3.3 Group method of data handling-based networks

The GMDH is a supervised algorithm used for modeling complex nonlinear systems [24]. It builds the training model to avoid the data overfitting problem and validates it by employing a multi-criteria objective function [31]. This is achieved by considering a quadratic polynomial relationship between the output and input elements so as to generate the minimum prediction error. The architecture of GMDH based model [12] has been presented in Fig. 1.

Let $D = \{d_1, d_2, \dots, d_n\}$ be the input set of n instances and $Y = \{y_1, y_2, \dots, y_i\}$ be the predictor set. For example, two inputs, say d_i and d_j , and a third-degree polynomial equation are combinedly used to represent a two node GMDH model [5], which can be depicted as follows:

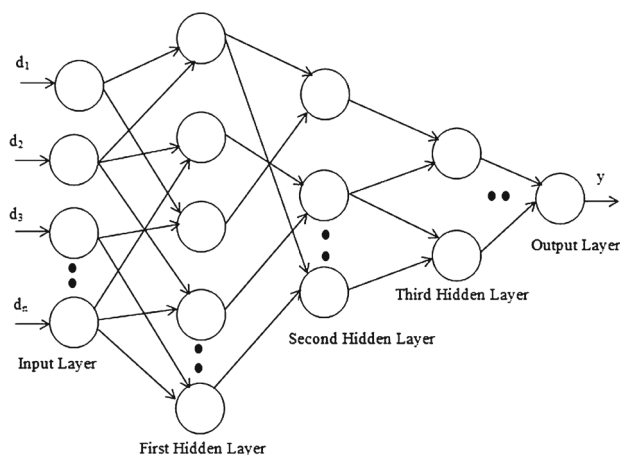


Fig. 1 Architecture of GMDH model

$$y = t_0 + t_1 d_i + t_2 d_j + t_3 d_i^2 + t_4 d_j^2 + t_5 d_i d_j + t_6 d_i^3 + t_7 d_j^3, \quad (2)$$

where y is the predictor variable of the node, while t_0 to t_7 represent the coefficients. The dataset is fed into the GMDH model through an input layer. Afterward, regressions of these inputs are computed and the best ones are extracted to form the first layer. Similarly, the second layer is constructed from the best nodes estimated from the regression between the first layer and the input layer values. The designing of GMDH network is completed if the condition for a particular layer's best neuron exceeds the stopping criterion and the polynomial expression of that neuron is used as the final output y . If not, the next layer is generated, and this process goes on. Finally, the optimum result y is produced with the minimum prediction error [24].

4 Proposed fraud detection model

In this work, initially, the proposed FDS builds subscriber calling profiles from the past call detail records (CDRs) by applying the GA-based FCM (GAFCM) clustering. During the fraud identification phase, a test call record is passed through the GAFCM clustering module which classifies the instances into three categories—genuine, fraudulent and suspicious. The records identified as legitimate and malicious are discarded and the suspicious points are further analyzed by GMDH for classification purpose. The proposed FDS depicted in Fig. 2 comprises two components that have been discussed in the following subsections.

1. Profile building.
2. Fraud identifier.

4.1 Profile building

The profile building component deals with the construction of user behavioral profiles by considering the following features:

$\langle user_id, call_type, call_dur, time_stm \rangle$,

where

- *user_id*: unique anonymized IMEI (International Mobile Equipment Identity) number.
- *call_type*: type of calls (local, national, international) made by the user. The values are converted to integers—local as 0, national as 1 and international as 2.
- *call_dur*: call duration (in seconds).
- *time_stm*: call time (hh:min:sec in 24-h format) and date (dd-mm-yyyy).

For instance, let (9, 1, 43, 04052005011530) be the call record of a subscriber, which describes that the subscriber having *user_id* = 9 has made a national call (*call_type* = 1) of *call_dur* = 43 s on date 04-05-2005 during time 01:15:30 (*time_stm* = 04052005011530). The profile building module comprises two sub-modules, namely, attribute filtration and GAFCM clustering for subscriber profile generation.

4.1.1 Attribute filtration

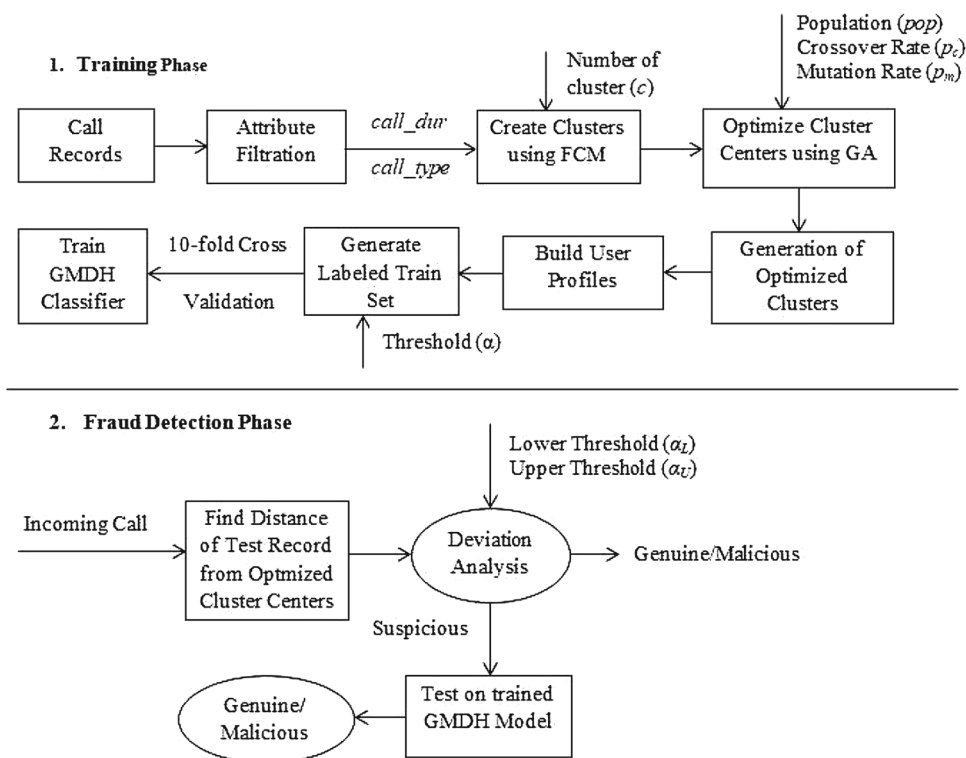
The raw dataset was preprocessed before the fraud detection process begins. Initially, the categorical attribute *call_type* was mapped into numerical values as the computation needed for cluster formation is based on integers. Secondly, the attribute values are normalized to [0, 1] range as the largely valued attributes affect the cluster performance. Finally, the features—*call_type* and *call_dur* have been chosen for user profile construction by the GAFCM clustering module.

4.1.2 GAFCM clustering

This sub-component takes two attributes—*call_dur* and *call_type* along with the cluster number (c) as input and groups them by applying the FCM algorithm. As the performance of FCM is susceptible toward the random initialization of cluster centers, the GA is used on the cluster centers for expanding and optimizing its search space globally, thus helping FCM to generate more robust clusters. The cluster number c was determined experimentally by computing some fuzzy performance indices as presented in Table 2 in Sect. 5.1.

To carry out the optimization procedure, the GA parameters—population size (pop), crossover rate (p_c), mutation

Fig. 2 Workflow of the proposed model



rate (p_m), genome length (l), and cluster center matrix (V) are initially set. The genome length (l) is the total count of features present in the train set, while the matrix V is of size $(c \times l)$. The GA tends to take more computation time for finding the best solution for a large population size over a large number of iterations [28]. Furthermore, a high p_c value leads to the generation of new chromosomes faster, while a low value declines the creation rate [19]. Similarly, a small p_m value increases the chance of infusing variability in a new population [19]. The functional parameters required for GA have been determined experimentally by finding out the minimum cost of the fitness function (Eq. 1) and are presented in Table 3 in Sect. 5.2.

For optimizing the cluster centers of FCM, we have encoded each variable of V matrix into some strings of binary numbers 0s and 1s using binary encoding [9] and updated the V matrix iteratively as follows [8]:

$$v_i = \frac{\sum_{j=1}^n u_{ij}^m \cdot d_j}{\sum_{j=1}^n u_{ij}^m}, \tag{3}$$

where the fuzzy weighting exponent is m , n is the total number of points present in the dataset and $U = [u_{ij}]$ is the fuzzy membership matrix. Similarly, the U matrix specified in Eq. (1) is updated in each iteration [8] as follows:

$$u_{ik} = 1 / \sum_{j=1}^c \left[\frac{B_{ik}(v_i, d_k)}{B_{jk}(v_j, d_k)} \right]^{1/(m-1)}$$

$$\text{for } 1 \leq i \leq c \text{ and } 1 \leq k \leq n, \tag{4}$$

where $B_{ik}(v_i, d_k)$ signifies any distance measure between the data instance d_k and cluster center v_i . On each iteration, these two matrices are updated according to Eqs. (3) and (4) in such a way that a minimum fitness function cost (i.e., Eq. 1) is achieved while producing the optimal clustering structure. The Euclidean distance of an instance is calculated from the optimized cluster centers as follows:

$$e = \sqrt{\sum_{i=1}^n (v_i - d_i)^2}, \tag{5}$$

where the Euclidean distance is e , cluster center is v_i and instance is d_i , while the total points present in the dataset is n . The FCM assigns the new calling instance in a cluster according to the fuzzy membership value. The membership value $\rightarrow 1$ denotes the high similarity toward a cluster, while membership value $\rightarrow 0$ indicates less similarity. The estimated distance is then compared with a threshold value (α) determined by the Tukey method for threshold detection [43]. For a dataset $D = \{d_1, d_2, \dots, d_n\}$, it first sorts them chronologically in ascending order and then categorizes into four quarters called Q_1 (1st quartile), Q_2 (2nd quartile) and Q_3 (3rd quartile). The threshold value estimated by the quartiles is expressed as follows:

$$\alpha = Q_3 + 3\|Q_3 - Q_1\|. \tag{6}$$

The call records are labeled as malicious for $e > \alpha$, while other points are marked as normal. These labeled calling instances are then subjected to the GMDH classifier for generating a trained model.

4.2 Fraud identifier

Upon receiving a new call record, this component detects the occurrences of fraudulent activities in two stages. The discrimination of genuine and fraudulent calls is done by using two thresholds, namely an upper threshold (α_U) and a lower threshold (α_L) for better classification and minimization of misclassified instances. In the first stage, after computing the Euclidean distance by using Eq. (5), it is compared with two threshold values α_L and α_U , respectively. The upper threshold (α_U) is determined by Eq. (6), while the lower threshold (α_L) is estimated by applying the Tukey method as expressed below:

$$\alpha_L = Q_1 - 3\|Q_3 - Q_1\|. \quad (7)$$

The segregation of the new call record is carried out as follows:

- If $e < \alpha_L$, then the call record is marked as legitimate.
- For $e > \alpha_U$, the calling instance is labeled as malicious and a confirmation is made by the service provider company from the corresponding subscriber regarding this event.
- If $\alpha_L \leq e \leq \alpha_U$, then the incoming call record is identified as suspicious and further investigation is done by the previously trained GMDH-based neural network model.

In the second stage, the GMDH model is employed for scrutinizing the suspicious call records and classifying them into genuine or fraudulent classes. Since GMDH being a supervised classifier, the legitimate and malicious instances generated from the clustering module are given to the GMDH for building a trained model. The tenfold cross-validation [35] is employed to train and validate the model. Initially, this method divides the train set into ten subsamples arbitrarily, out of which nine subsamples are combinedly used for training and the remaining one subsample is taken for validation. This process continues ten times to generate ten different trained GMDH models. The validation set then is employed on these models to find out the respective misclassification rate. The model generating the lowest misclassification rate is finally selected as the best GMDH model. When the suspicious call instances are given to the validated model, it makes the final decision (genuine/malicious) by utilizing Eq. (2).

Table 1 compiles a list of acronyms with their description used in the current model.

Table 1 List of acronyms used in the proposed system

Acronyms	Description
GA	Genetic algorithm
PC	Partition coefficient
PE	Partition entropy
TP	True positive
TN	True negative
FP	False positive
FN	False negative
CDR	Call detail record
FCM	Fuzzy C-means
FDS	Fraud detection system
FPI	Fuzziness performance index
NCE	Normalized classification entropy
ICDR	Internal cluster dispersion rate
IMEI	International mobile equipment identity
GMDH	Group method of data handling
GAFCM	Genetic algorithm-based fuzzy C-means

5 Results and discussion

Experimentation was conducted on a 2.40 GHz i5 CPU system and the proposed model was implemented in MATLAB 8.3. The performance of our proposed system was tested on a real-world call dataset. Several tests were done to determine optimal parameter values required for FCM and GA, respectively. After the parameter estimation was over, the effectiveness of the current system was evaluated.

In this work, we have used the Reality Mining dataset [13] that contains call and message details and other information of 106 subscribers gathered during Sept. 2004 to April 2005 time period. This dataset has successfully been analyzed for studying the changes in behavioral patterns of people [17], the discovery of social relationships [14] as well as for classification purpose [16]. The data preprocessing procedure is then followed to handle the raw dataset. Afterward, we applied GAFCM clustering to generate subscriber's calling behavioral profiles. The dataset containing 1,28,541 calling instances are segregated into train and test sets of size 1,15,687 and 12,854 records, respectively.

5.1 FCM parameter estimation

Experimentation is done to determine the required cluster number (c) for effective FCM clustering. Two fuzzy metrics—partition entropy (PE) and partition coefficient (PC) are considered to compute the optimal cluster number [44]. The PC measures the average amount of membership present in between any two fuzzy subsets that can be

Table 2 Determination of optimum cluster number

c	PC	PE	FPI	NCE
2	0.8936	0.2019	0.2128	0.2912
3	0.9932	0.0162	0.0103	0.0152
4	0.9763	0.0465	0.0315	0.0336
5	0.9627	0.0730	0.0467	0.0454

The optimal cluster number c^+ has been highlighted in bold for better visualization

expressed as:

$$PC = 1/n \sum_{i=1}^c \sum_{j=1}^n u_{ij}^2, \tag{8}$$

where the cluster number is c , the dataset on which clustering has to be performed contains n instances and $U = [u_{ij}]$ refers to the membership matrix. The value of c^+ is found at $\max_{2 \leq c \leq n-1} PC$. Similarly, the PE estimates the amount of fuzziness present in matrix U , which can be described as:

$$PE = -1/n \sum_{i=1}^c \sum_{j=1}^n u_{ij} \log u_{ij}. \tag{9}$$

The value of (c^+) can be derived from $\min_{2 \leq c \leq n-1} PE$. Moreover, two other cluster validity measures [6] were used—fuzziness performance index (FPI) and normalized classification entropy (NCE). They evaluate the degree of separation between clusters. The FPI quantifies the amount of shared membership between different classes, whereas NCE measures how many clusters are most appropriate for an efficient grouping. The FPI is expressed as:

$$FPI = 1 - \frac{(c * PC - 1)}{c - 1}, \tag{10}$$

where PC is the partition coefficient as computed in Eq. (8). Likewise, the NCE can be computed as follows:

$$NCE = \frac{PE}{\log n}, \tag{11}$$

where PE indicates the partition entropy as shown in Eq. (9). More distinct partitions can be found for smaller values of FPI and NCE [6]. We have considered the Bezdek’s suggestion: $c_{\min} = 2$ for selecting the best value of cluster number [7]. The optimal cluster number c^+ has been highlighted in bold in Table 2 for better visualization. For $c = 3$, both the $PC = 0.9932$ (maximum) and $PE = 0.0162$ (minimum), while both FPI and NCE produce the least value. Hence, we have chosen $c^+ = 3$ for FCM clustering in the rest of our experiments.

5.2 Determination of GA parameters

In this subsection, several tests are conducted to estimate the optimum GA parameter combination. The effectiveness of GA greatly relies on three parameters—crossover rate (p_c), population size (pop) and mutation rate (p_m) as discussed in Sect. 3.1. The parameters giving the least cost of objective (fitness) function has been chosen as the optimum ones, since lower cost value gives better performance [28]. Table 3 presents the cost function value with respect to different combinations of the aforementioned three GA parameters. The pop values are taken in the range of [10, 100] in increasing steps of 10, while p_c values are ranged from 0.6 to 1.0 by incremental steps of 0.1. Likewise, the p_m values have been varied in between 0.02 to 0.1 by adding 0.02 to each.

It is clearly evident from the table that the GA produces the lowest cost= $9.9543e-5$ at $pop = 50$, $p_c = 0.8$ and $p_m = 0.02$. Hence, we have selected these parameter values. Moreover, the number of iterations required for computation of the GA optimization function has a greater effect on the computational time. Table 4 presents the performance of the GA over different iterations, starting from 100 to 1000 in increasing steps of 100, with respect to the time measured in seconds. It is visible from the table that the time increases proportionally with the iteration number. Therefore, the number of iterations = 100 was selected for generating the least computational time.

5.3 Performance of the GAFCM clustering module

After determining the optimal parameters required for GA and FCM, we then performed some tests on optimizing the cluster center by applying GA on c^+ . Table 5 presents the GAFCM clustering output with respect to the FPI and NCE values. It has been observed from the table that both FPI and NCE values of Run 6 are minimum. Hence, we have chosen the center (c) of Run 6 as a result of optimized clustering.

Figure 3 depicts the spread of GAFCM objective function corresponding to 100 iterations. It has been seen by analyzing the figure that around the 24th iteration, the fitness function attains the optimal value and after that it remains constant for higher iteration steps.

After the clustering process is over, the Euclidean distance (e) of the train points with respect to the optimized cluster centers are computed by using Eq. (5). The values of the first quartile $Q_1 = 0.1384$ and the third quartile $Q_3 = 0.3836$ are generated by the Tukey method. Finally, the threshold α is found to be 1.1192 by utilizing Eq. (6). This leads to the generation of genuine samples of size 1,01,977 and forged instances of 13,710 records from the training set of 1,15,687 rows, which are then used for training the GMDH classifier.

During the fraud detection phase, when the test set consisting of 12,854 call records are given to the clustering

Table 3 Determination of optimum GA parameters

Population size	Crossover rate (p_c)				
	0.6	0.7	0.8	0.9	1.0
(a) Mutation rate $p_m = 0.02$					
10	1.2241e-4	9.1120e-5	8.7375e-5	1.0013e-4	7.4848e-5
20	2.0229e-4	2.8823e-4	1.7231e-4	2.1835e-4	2.5681e-4
30	1.7386e-4	1.7333e-4	1.8558e-4	1.6626e-4	1.9567e-4
40	2.0356e-4	1.5936e-4	1.9970e-4	1.1876e-4	1.1120e-4
50	1.6784e-4	1.5227e-4	9.9543e-5	1.7375e-4	3.2242e-5
60	2.0936e-4	1.7686e-4	1.4117e-4	1.1524e-4	1.6922e-4
70	1.1488e-4	1.2654e-4	1.3303e-4	1.3666e-4	1.3098e-4
80	1.1308e-4	1.5984e-4	1.4552e-4	1.4967e-4	1.3854e-4
90	1.0229e-4	9.7952e-5	1.5172e-4	1.3622e-4	1.1622e-4
100	1.4176e-4	1.6093e-4	1.1116e-4	1.1670e-4	1.3501e-4
(b) Mutation rate $p_m = 0.04$					
10	1.0623e-4	1.1040e-4	1.4670e-4	1.2786e-4	1.0915e-4
20	2.1789e-4	1.9273e-4	1.7092e-4	1.8512e-4	2.1070e-4
30	1.3316e-4	1.5556e-4	1.6467e-4	2.1287e-4	1.4896e-4
40	1.7210e-4	1.7834e-4	1.2551e-4	1.6770e-4	1.6571e-4
50	1.5416e-4	1.2833e-4	1.3447e-4	1.4004e-4	1.7257e-4
60	1.3909e-4	1.1789e-4	1.2730e-4	1.6177e-4	1.5857e-4
70	1.1908e-4	1.2897e-4	1.1089e-4	1.4511e-4	1.3540e-4
80	9.7051e-5	1.3285e-4	1.1971e-4	1.3082e-4	1.1490e-4
90	1.6273e-4	1.2045e-4	1.1890e-4	1.5493e-4	2.3316e-4
100	1.2080e-4	1.1736e-4	1.1428e-4	1.4202e-4	1.3001e-4
(c) Mutation rate $p_m = 0.06$					
10	1.0981e-4	1.1270e-4	7.6436e-5	9.1738e-5	7.0817e-5
20	5.5119e-4	2.2109e-4	2.2052e-4	2.1293e-4	2.4559e-4
30	1.6402e-4	2.6194e-4	1.9561e-4	1.1738e-4	2.2425e-4
40	1.3831e-4	1.0338e-4	1.4209e-4	1.4470e-4	1.8840e-4
50	1.2894e-4	1.3299e-4	1.3600e-4	1.2741e-4	1.5761e-4
60	8.1917e-5	1.0774e-4	1.5515e-4	1.4271e-4	1.3475e-4
70	1.4219e-4	1.3142e-4	1.2636e-4	1.1203e-4	1.3700e-4
80	1.5437e-4	1.5508e-4	1.0873e-4	1.1528e-4	1.3167e-4
90	1.0278e-4	2.0047e-4	1.2380e-4	1.2260e-4	1.2819e-4
100	1.3655e-4	1.1291e-4	1.2759e-4	1.2380e-4	1.2935e-4
(d) Mutation rate $p_m = 0.08$					
10	1.0217e-4	9.4888e-5	8.2749e-5	1.0683e-4	8.5852e-5
20	1.8642e-4	2.2233e-4	2.6801e-4	1.9226e-4	2.4341e-4
30	1.6017e-4	1.7014e-4	1.9854e-4	1.6306e-4	2.4330e-4
40	1.2729e-4	3.1875e-5	1.2345e-4	2.0240e-4	2.4610e-4
50	1.6665e-4	1.2831e-4	1.4319e-4	1.3862e-4	1.3328e-4
60	1.3782e-4	1.4065e-4	1.2749e-4	2.0442e-4	1.8755e-4
70	1.2015e-4	1.4329e-4	1.7243e-4	1.3715e-4	1.3461e-4
80	1.2638e-4	1.4096e-4	1.2233e-4	1.0945e-4	1.5220e-4
90	1.2116e-4	1.2042e-4	1.3045e-4	1.3360e-4	1.6071e-4
100	1.3562e-4	2.4065e-4	1.2270e-4	1.1687e-4	1.6295e-4

Table 3 continued

Population size	Crossover rate (p_c)				
	0.6	0.7	0.8	0.9	1.0
(e) Mutation rate $p_m = 0.1$					
10	9.5195e-5	1.0025e-4	7.9131e-5	7.4710e-5	1.5384e-4
20	2.1180e-4	2.3765e-4	2.1866e-4	1.7400e-4	2.4038e-4
30	1.5675e-4	1.5581e-4	1.5131e-4	1.4134e-4	2.2912e-4
40	1.6878e-4	1.2117e-4	2.7142e-4	4.4644e-5	1.7701e-4
50	1.3763e-4	1.7839e-4	1.4275e-4	1.2755e-4	3.0468e-5
60	1.2393e-4	1.3136e-4	1.6429e-3	3.1390e-5	1.6380e-4
70	3.6001e-5	1.1482e-4	1.3610e-4	1.2515e-4	2.8997e-5
80	1.3932e-4	1.3832e-4	1.7499e-4	1.3806e-4	3.1234e-5
90	1.0251e-4	1.2955e-4	1.1785e-4	1.1623e-4	3.7351e-5
100	3.3317e-5	1.0070e-4	9.2586e-5	1.6864e-4	3.1922e-5

The bold letter has been used to highlight the best values that are obtained during the experimentation

Table 4 Determination of optimal iteration number for GA

Number of iterations	100	200	300	400	500	600	700	800	900	1000
Computational time (s)	122	185	248	290	317	368	425	490	531	601

The bold letter has been used to highlight the best values that are obtained during the experimentation

Table 5 Results produced by GAFCM with different performance indices

Run	1	2	3	4	5	6	7
v_1	0.4603	0.5757	0.3809	0.4048	0.6746	0.2857	0.5159
v_2	0.0952	0.6508	0.9445	0.2064	0.7222	0.5159	0.3492
v_3	0.0634	0.5000	0.1032	0.5953	0.0953	0.6014	0.0794
FPI	-2.36e+04	-2.54e+04	-3.29e+04	-2.73e+04	-3.06e+04	-4.41e+04	-3.84e+04
NCE	3.41e+04	3.33e+04	2.54e+04	3.07e+04	2.83e+04	1.61e+04	2.17e+04

The bold letter has been used to highlight the best values that are obtained during the experimentation

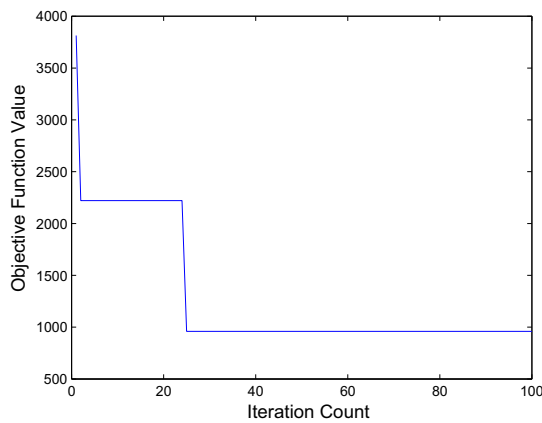


Fig. 3 Fitness function optimization over 100 iterations

module, the Euclidean distances are computed from the optimized cluster centers by utilizing Eq. (5). These distances are then compared with a lower threshold α_L and an upper limit α_U for discriminating the test instances into

genuine, malicious or suspicious classes. The quartile values needed for calculating the boundary values are found to be $Q_1 = 0.3809$ and $Q_3 = 0.4009$. This produces the threshold values $\alpha_U = 0.4609$ and $\alpha_L = 0.3209$ estimated by Eq. (6) and Eq. (7), respectively. The test set is then segregated into 9437 genuine records, 2128 suspicious samples and 1289 fraudulent instances in the first stage.

5.4 Performance of the model

The performance of the whole system has been presented in this section after identifying the fraudulent activities by the GMDH. As mentioned in Sect. 3.3, parameters required for effective performance of GMDH are determined automatically so as to minimize the misclassification rate. The number of layers required for the functioning of GMDH is found to be at 3 with 15 neurons in each layer.

The following metrics—Accuracy, Sensitivity, Precision, Specificity, and F -Score have been considered to estimate the efficiency of the suggested FDS. Sensitivity counts the frac-

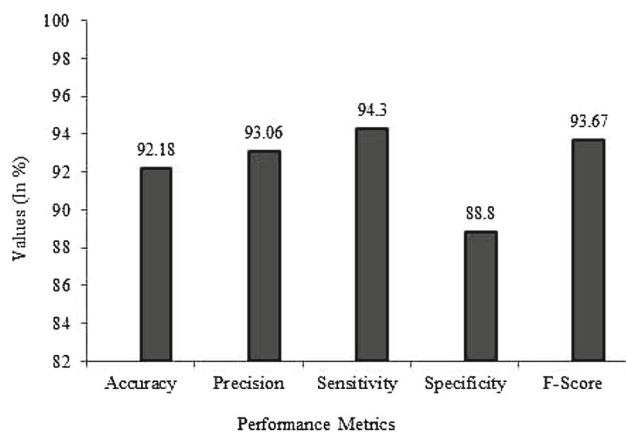


Fig. 4 Performance analysis of the model

tion of truly genuine instances that are precisely detected by the system. Specificity denotes the ratio of correctly detected true positive and true negative samples. Accuracy estimates the correctness of the model, and Precision measures the amount of accurate classification done by the model, while *F*-Score is determined from Precision and Sensitivity.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (12)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \quad (13)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (14)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (15)$$

$$F\text{-Score} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}}, \quad (16)$$

where FN is the false negative, TP denotes true positive, FP signifies false positive and TN refers to true negative.

The efficacy of the proposed system corresponding to the above-mentioned performance metrics has been illustrated in Fig. 4. It is clearly visible from the figure that the system is capable of detecting fraudulent calls efficiently by keeping the Specificity value high (i.e., low false alarm rate).

5.5 Comparative performance study

A comparative analysis of the proposed FDS has been done in this section with two other mobile phone fraud detection approaches found in the literature [21,40]. Experiments are done on the Reality Mining dataset [13] while considering the above-mentioned performance metrics.

In paper [21], the authors have generated five different user profiles for modeling the subscriber's behavior based on daily and weekly call analysis. They have analyzed these profiles for identification of fraudulent behaviors by employing K-

means clustering (hereby called *KC_FDS*) and hierarchical agglomerative clustering (hereby called *HAC_FDS*) on them individually. Here, in this paper, the authors have termed these profiles as follows:

- *Profile 1* the weekly behavior of a user comprising the standard deviation and mean of the calls and their duration, maximum call duration, maximum call cost and a maximum number of calls.
- *Profile 2* detailed daily behavior of a user based on the combination of types of calls—national (*Nat*), local (*Loc*) and international (*Int*) and time of call—work (*w*), afternoon (*a*) and night (*n*).
- *Profile 3* accumulated per day behavior representing the number of calls made along with their duration based on the type of calls (*Loc*, *Nat*, *Int*).
- *Profile 2w* the weekly call analysis of a subscriber based on *Profile 2* and
- *Profile 3w* accumulated weekly behavior based on *Profile 3*.

In this work, we have considered four profiles—Profile 2, Profile 3, Profile 2w and Profile 3w for comparison as the cost attribute values required for Profile 1 is unavailable in the dataset [13]. These four profiles are generated from our dataset according to the steps suggested in [21]. The nomenclature for all the profiles are also kept same as that of [21] for clear understanding. Figures 5 and 6 present the Profile 2 and Profile 3, respectively.

After these four subscriber profiles were generated, the fraud identification procedure of the proposed model was conducted for each profile by keeping the model parameters same in all cases. Table 6 presents the values of Sensitivity and Specificity, measured in %, obtained in case of our proposed approach, *KC_FDS* and *HAC_FDS* experimented on the same dataset [13].

It is clearly depicted from Table 6 that our proposed approach produces the highest Sensitivity = 89.36% than that of *HAC_FDS* and *KC_FDS* on all profiles. However, the proposed FDS exhibits optimal performance in Profile 3 by claiming maximum Specificity = 88.46% (i.e., least false acceptance rate). It is to be noted that gaining high Sensitivity and Specificity is desirable for achieving effective classification result [40]. Similarly, Table 7 gives an insight into the comparative performance of our approach, *KC_FDS* and *HAC_FDS* in terms of Accuracy, Precision and *F*-Score measured in %. It is observed from the table that our FDS outperforms the other two approaches in all profiles by displaying better results in terms of Precision, *F*-Score and Accuracy values. Moreover, by attaining the highest Precision = 93.33%, we conclude that the current model captures the subscriber's behavior more accurately in all profiles than *KC_FDS* and *HAC_FDS*.

Fig. 5 Profile 2 of user

<i>Loc_calls_w</i>	<i>Loc_dur_w</i>	<i>Loc_calls_a</i>	<i>Loc_dur_a</i>	<i>Loc_calls_n</i>	<i>Loc_dur_n</i>
<i>Nat_calls_w</i>	<i>Nat_dur_w</i>	<i>Nat_calls_a</i>	<i>Nat_dur_a</i>	<i>Nat_calls_n</i>	<i>Nat_dur_n</i>
<i>Int_calls_w</i>	<i>Int_dur_w</i>	<i>Int_calls_a</i>	<i>Int_dur_a</i>	<i>Int_calls_n</i>	<i>Int_dur_n</i>

Fig. 6 Profile 3 of user

<i>Loc_calls</i>	<i>Loc_dur</i>	<i>Nat_calls</i>	<i>Nat_dur</i>	<i>Int_calls</i>	<i>Int_dur</i>
------------------	----------------	------------------	----------------	------------------	----------------

Table 6 Performance analysis of our approach, KC_FDS and HAC_FDS

Profile	Our approach		KC_FDS		HAC_FDS	
	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity
Profile 2	89.36	77.78	76.19	78.01	84.31	78.90
Profile 3	87.50	88.46	75.70	76.98	78.38	80.00
Profile 2w	85.71	75.00	84.21	41.18	51.72	26.32
Profile 3w	87.50	66.67	78.95	47.06	85.36	31.58

The bold letter has been used to highlight the best values that are obtained during the experimentation

In another work [40], the authors have suggested using FCM clustering and support vector machine (SVM) for identification of fraudulent behavior through user profile building and hence named as *FCMSVM_FDS*. Initially, the past call records of a subscriber are given as input to FCM and calling behavioral profiles are generated for each user via cluster formation. These behavioral patterns are then passed through the SVM classifier model [11] for training and classification purposes. Any discrepancy or inconsistency found in the current behavior from the user profile indicates a fraud.

To compare GAFCM and FCM, another metric known as ICDR (internal cluster dispersion rate) is used that determines the amount of scattered instances inside a clustering structure [15]. The ICDR can be mathematically expressed as:

$$ICDR = 1 - \frac{\sum_{i=1}^c \text{dist}_{i0}^2}{\sum_{j=1}^n \text{dist}_j^2}, \tag{17}$$

where dist_{i0} refers to the Euclidean distance of i th cluster center with the mean of the whole dataset, while dist_j signifies the Euclidean distance of j th point with the mean of

Table 8 Performance comparison of clustering techniques

Clustering techniques	ICDR
FCM	0.0489
GAFCM	0.0412

The bold letter has been used to highlight the best values that are obtained during the experimentation

the overall dataset. The cluster number is c and n denotes the instances present in the dataset. The lesser ICDR signifies better clustering since it exhibits smaller intra-class cluster dispersion [15].

Table 8 presents a comparative analysis of the performance of FCM and GAFCM clustering techniques with respect to ICDR value. From the table, it is quite clear that the GAFCM generates better clusters with the lowest $ICDR = 0.0412$ as compared to FCM.

Furthermore, a comparative analysis of our proposed system, *KC_FDS*, *HAC_FDS* and *FCMSVM_FDS*, has been given in Table 9 with respect to (w.r.t) the fraud detection

Table 7 Performance metrics of our approach, KC_FDS and HAC_FDS

Profile	Our approach			KC_FDS			HAC_FDS		
	Accuracy	Precision	F Score	Accuracy	Precision	F Score	Accuracy	Precision	F Score
Profile 2	85.14	87.50	88.42	77.24	72.07	74.07	79.27	74.78	77.13
Profile 3	87.84	93.33	90.32	76.42	71.68	77.14	79.27	76.32	77.33
Profile 2w	81.82	85.71	85.71	63.89	61.54	71.11	55.56	51.72	65.22
Profile 3w	81.82	87.50	87.50	63.89	62.50	47.06	55.56	51.85	63.64

The bold letter has been used to highlight the best values that are obtained during the experimentation

Table 9 Comparison of various approaches w.r.t fraud detection time

Approaches	Fraud detection time (s)
Proposed approach	4.71
KC_FDS	8.22
HAC_FDS	11.37
FCMSVM_FDS	4.12

The bold letter has been used to highlight the best values that are obtained during the experimentation

time measured in seconds. It is observed that our proposed FDS is able to identify the fraudulent activities much faster than *KC_FDS* and *HAC_FDS*, i.e., time = 4.12 s. However, the *FCMSVM_FDS* outperforms all the approaches in terms of fraud detection time by taking only 4.12 s.

6 Conclusions

The current paper suggests a novel mobile phone fraud detection approach proceeding in two phases—training and fraud detection. In the training phase, a GA-based FCM clustering has been employed on the subscriber's historical calls for effectively modeling the calling patterns. The FCM has been used for clustering, while GA is applied on FCM for optimizing the cluster centers. For each new incoming calling instance, the fraud detection process is carried out by passing it through GAFCM that calculates Euclidean distance from the optimized cluster centers. The incoming call is categorized into either of any genuine, malicious or suspicious classes after being compared with two threshold values determined by the Tukey method. The data points marked as genuine and forged are not processed further, while the suspicious samples are additionally analyzed and verified by a previously trained GMDH supervised classifier.

Extensive experimentation was done for evaluating the efficacy of the proposed system on a real-world large-scale unlabeled Reality Mining dataset. The tenfold cross-validation is employed throughout the procedure for segregating the dataset into train and test set as well as training and validation of the system. Initially, experiments were conducted for finding out the optimal parameter values required for FCM and GA. Several tests were further carried out for the generation of optimized cluster centers by applying GAFCM. The whole dataset consisting of 1,28,541 records is divided into train and test sets of size 1,15,687 and 12,854 samples, respectively. After employing GAFCM on the train set, 1,01,977 genuine and 13,710 fraud labeled samples were generated, which were then used for training the GMDH model for learning the user behaviors.

For evaluating the performance of the model, a test set having 12,854 records was applied to the proposed system.

After applying GAFCM clustering on the test samples, 9437 records were found to be genuine, 2128 points as suspicious and 1289 as fraudulent instances. In the learning phase, the 2128 suspicious data points were additionally verified by the trained GMDH model for final classification. It is found that the proposed system yielded 94.30% Sensitivity and 88.80% Specificity with a Precision of 93.06%. Besides, the results obtained from the comparative performance analysis with a recent mobile phone FDS and one of our earlier work clearly exhibit the superiority of the current model.

Acknowledgements The authors are extremely thankful to the Department of Computer Science and Engineering, Veer Surendra Sai University of Technology, Burla, India, for making this research successful.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Abdel-Aal, R.: Gmdh-based feature ranking and selection for improved classification of medical data. *J. Biomed. Inf.* **38**(6), 456–468 (2005)
2. Adhikari, S.K., Sing, J.K., Basu, D.K., Nasipuri, M.: Conditional spatial fuzzy c-means clustering algorithm for segmentation of mri images. *Appl. Soft Comput.* **34**, 758–769 (2015)
3. Agarwal, A.: Abductive networks for two-group classification: a comparison with neural networks. *J. Appl. Bus. Res. (JABR)* **15**(2), 1–12 (2011)
4. Bäck, T., Schwefel, H.P.: An overview of evolutionary algorithms for parameter optimization. *Evol. Comput.* **1**(1), 1–23 (1993)
5. Baig, Z.A., Sait, S.M., Shaheen, A.: Gmdh-based networks for intelligent intrusion detection. *Eng. Appl. Artif. Intel.* **26**(7), 1731–1740 (2013)
6. Bezdek, J.C., Coray, C., Gunderson, R., Watson, J.: Detection and characterization of cluster substructure i. linear structure: fuzzy c-lines. *SIAM J. Appl. Math.* **40**(2), 339–357 (1981)
7. Bezdek, J.C., Ehrlich, R., Full, W.: Fcm: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **10**(2), 191–203 (1984)
8. Bezdek, J.C., Hathaway, R.J.: Optimization of fuzzy clustering criteria using genetic algorithms. In: *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pp. 589–594. IEEE (1994)
9. Boussaid, I., Lepagnot, J., Siarry, P.: A survey on optimization metaheuristics. *Inf. Sci.* **237**, 82–117 (2013)
10. Burge, P.: Novel techniques for profiling and fraud detection in mobile telecommunications. *Bus. Appl. Neural Netw* 113–139 (2000)
11. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995)
12. Demertzis, K., Iliadis, L., Avramidis, S., El-Kassaby, Y.A.: Machine learning use in predicting interior spruce wood density utilizing progeny test information. *Neural Comput. Appl.* **28**(3), 505–519 (2017)
13. Eagle, N., Pentland, A.: Reality mining: sensing complex social systems. *Person. Ubiquit. Comput.* **10**(4), 255–268 (2006)

14. Eagle, N., Pentland, A.S.: Eigenbehaviors: identifying structure in routine. *Behav. Ecol. Sociobiol.* **63**(7), 1057–1066 (2009)
15. Everitt, B., Landau, S., Leese, M., Stahl, D.: *Cluster Analysis*, p. 330. Wiley, New York (2011)
16. Ferrari, L., Mamei, M.: Classification and prediction of whereabouts patterns from the reality mining dataset. *Pervas. Mob. Comput.* **9**(4), 516–527 (2013)
17. Ficek, M., Kencl, L.: Spatial extension of the reality mining dataset. In: *Mobile Adhoc and Sensor Systems (MASS)*, 2010 IEEE 7th International Conference on, pp. 666–673. IEEE (2010)
18. Gosset, P., Hyland, M.: Classification, detection and prosecution of fraud in mobile networks. In: *Proceedings of ACTS Mobile Summit, Sorrento, Italy* (1999)
19. Grefenstette, J.J.: Optimization of control parameters for genetic algorithms. *IEEE Trans. Syst. Man Cybernet.* **16**(1), 122–128 (1986)
20. Hilas, C.S., Kazarlis, S.A., Rekanos, I.T., Mastorocostas, P.A.: A genetic programming approach to telecommunications fraud detection and classification. In: *Proc. 2014 Int. Conf. Circuits, Syst. Signal Process. Commun. Comput.*, pp. 77–83 (2014)
21. Hilas, C.S., Mastorocostas, P.A., Rekanos, I.T.: Clustering of telecommunications user profiles for fraud detection and security enhancement in large corporate networks: a case study. *Appl. Math. Inf. Sci.* **9**(4), 1709 (2015)
22. Holland, J.H.: *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Michigan Press, USA (1975)
23. Howell, J.: 2015 global fraud loss survey by cfca (communications fraud control association). <http://cfca.org/fraudlosssurvey/2015.pdf> (2016). Accessed: 04 Aug 2017
24. Ivakhnenko, A.: Heuristic self-organization in problems of engineering cybernetics. *Automatica* **6**(2), 207–219 (1970)
25. Klawonn, F., Keller, A.: Fuzzy clustering based on modified distance measures. In: *International Symposium on Intelligent Data Analysis*, pp. 291–301. Springer (1999)
26. Ko, M.M., Thwin, M.M.S.: Anomalous behavior detection in mobile network. In: *Genetic and Evolutionary Computing*, pp. 147–155. Springer (2015)
27. Kosmides, M.: Telephone fraud on rise in UK, study finds. <http://www.counter-fraud.com/fraud-types-n-z/telecoms-fraud/telephone-fraud-on-rise-in-uk-study-finds--1.htm> (2014). Accessed 30 Jan 2016
28. Koumousis, V.K., Katsaras, C.P.: A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. *IEEE Trans. Evol. Comput.* **10**(1), 19–28 (2006)
29. Li, T., Zhang, L., Lu, W., Hou, H., Liu, X., Pedrycz, W., Zhong, C.: Interval kernel fuzzy c-means clustering of incomplete data. *Neurocomputing* **237**, 316–331 (2017)
30. Ludwig, S.A.: Mapreduce-based fuzzy c-means clustering algorithm: implementation and scalability. *Int J. Mach. Learn. Cybernet.* **6**(6), 923–934 (2015)
31. Mehra, R.: Group method of data handling (gmdh): review and experience. In: *Decision and Control including the 16th Symposium on Adaptive Processes and A Special Symposium on Fuzzy Set Theory and Applications*, 1977 IEEE Conference on, pp. 29–34. IEEE (1977)
32. Olszewski, D.: A probabilistic approach to fraud detection in telecommunications. *Knowl. Based Syst.* **26**, 246–258 (2012)
33. Olszewski, D.: Fraud detection using self-organizing map visualizing the user profiles. *Knowl. Based Syst.* **70**, 324–334 (2014)
34. Ravisanakar, P., Ravi, V.: Financial distress prediction in banks using group method of data handling neural network, counter propagation neural network and fuzzy artmap. *Knowl. Based Syst.* **23**(8), 823–831 (2010)
35. Refaeilzadeh, P., Tang, L., Liu, H.: Cross-validation. In: *Encyclopedia of database systems*, pp. 532–538. Springer (2009)
36. Sharma, A., Onwubolu, G.C.: Intrusion detection system using hybrid differential evolution and group method of data handling approach. In: *2nd International Conference on Inductive Modelling Proceedings*, pp. 255–262. International Research and Training Center for Information Technologies and Systems (2008)
37. Son, L.H.: A novel kernel fuzzy clustering algorithm for geodemographic analysis. *Inf. Sci. Int. J.* **317**(C), 202–223 (2015)
38. Stokes, R.: Telecom fraud losses to top US \$46bn in 2013. <http://www.counter-fraud.com/fraud-types-n-z/telecoms-fraud/telecom-fraud-losses-to-top-us46bn-in-2013-93232.htm> (2013). Accessed 30 Jan 2016
39. Subudhi, S., Panigrahi, S.: Quarter-sphere support vector machine for fraud detection in mobile telecommunication networks. *Proc. Comput. Sci.* **48**, 353–359 (2015)
40. Subudhi, S., Panigrahi, S.: Use of fuzzy clustering and support vector machine for detecting fraud in mobile telecommunication networks. *Int. J. Secur. Netw.* **11**(1–2), 3–11 (2016)
41. Subudhi, S., Panigrahi, S., Behera, T.K.: Detection of mobile phone fraud using possibilistic fuzzy c-means clustering and hidden markov model. *Int. J. Synth. Emot. (IJSE)* **7**(2), 23–44 (2016)
42. Thong, P.H., et al.: Picture fuzzy clustering: a new computational intelligence method. *Soft Comput.* **20**(9), 3549–3562 (2016)
43. Tukey, J.W.: *Exploratory data analysis*. Addison-Wesley Series in Behavioral Science: Quantitative Methods. Addison-Wesley, Reading (1977)
44. Wang, W., Zhang, Y.: On fuzzy cluster validity indices. *Fuzzy Sets Syst.* **158**(19), 2095–2117 (2007)
45. Witczak, M., Korbicz, J., Mrugalski, M., Patton, R.J.: A gmdh neural network-based approach to robust fault diagnosis: application to the damadics benchmark problem. *Control Eng. Pract.* **14**(6), 671–683 (2006)
46. Yang, M.S., Nataliani, Y.: Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters. *Pattern Recogn.* **71**, 45–59 (2017)
47. Zheng, Y., Jeon, B., Xu, D., Wu, Q., Zhang, H.: Image segmentation by generalized hierarchical fuzzy c-means algorithm. *J. Intel. Fuzzy Syst.* **28**(2), 961–973 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.