International Journal of Industrial Electronics, Control and Optimization © 2022 IECO
Vol. 5, No. 1, pp. 63-76, Jan (2022)

A Novel Meta-heuristic Framework for Solving Power Theft Detection Problem: Cheetah Optimization Algorithm

Hassan Ghaedi[1], Seyed Reza Kamel Tabbakh[2,†] , and Reza Ghaemi[3]
[1] Department of Computer, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran
[2] Department of Computer, Mashhad Branch, Islamic Azad University, Mashhad, Iran
[3] Department of Computer, Quchan Branch, Islamic Azad University, Quchan, Iran

ABSTRACT

In this paper, a two-level stacking technique with feature selection is used to detect power theft. The first level of this technique uses base classifiers such as support vector machine (SVM), naïve Bayes (NB), and AdaBoost selected by evaluating the F-score and diversity criteria. The appropriate features of the base classifiers are selected using a new feature selection algorithm based on the cheetah optimization algorithm (CHOA). This algorithm ensures diversification and intensification in each step of running by adjusting the Attention parameter of the cheetahs. In the second level, a single-layer perceptron (SLP) model is used to obtain the weight of the base classifiers and combine their predictions. The proposed framework is evaluated on the Irish Social Science Data Archive (ISSDA) dataset, and MATLAB R2020b is used for simulation and evaluation. The results of the accuracy, recall, precision, and F-score, specificity, and receiver operating characteristic (ROC) criteria indicated the high efficiency of the proposed framework in detecting power theft.

Article Info

Keywords:
Cheetah optimization algorithm, Classification, Feature selection, Machine learning, Power theft detection
Article History:
Received 2021-08-13
Accepted 2021-12-06

I. INTRODUCTION

Nowadays, electricity theft is one of the major challenges faced by power companies all over the world. Early detection of power theft can minimize the financial losses of these companies. Various methods have been used to detect power theft. Many researchers have used data mining techniques and modeling by using customers' consumption patterns for theft detection.

Passos Júnior et al. [1] used the optimum-path forest (OPF) clustering algorithm to identify normal and abnormal customer profiles in their research. They also provided a model for identifying non-technical losses (NTL) when there was little or no information about unauthorized customers. Jokar et al. [2] provided a consumption pattern-based power theft detector that uses distribution transformer meters to list areas with a high probability of theft in a short time and monitors abnormal uses to identify suspicious customers. They also used clustering and classification techniques to detect theft. In their review, Ahmad et al. [3] focused on different models for identifying non-technical losses. In their study, several methods of power theft detection were examined and discussed such as the SVM [4] model, optimum-path forest clustering process, real-time estimation techniques, SVM model with a genetic algorithm, decision trees (DT), NB network [5], etc, and their advantages were mentioned. Zheng et al. [6] presented a density-based power theft detection method to detect abnormal consumption patterns and used several malicious types for validation testing. The method was compared with several clustering algorithms. Xiao and Ai [7] developed a data-driven theft

[†]Corresponding Author: Rezakamel@ieee.org
Tel: +98-9151237429, Department of Computer, Mashhad Branch, Islamic Azad University, Mashhad

detector that uses random matrix theory with a wide range of smart meters. The use of matrix theory as a data source is one of the key steps in this method, in which areas with a high probability of theft and customers suspected of theft are quickly listed. Zheng et al. [8] proposed a method of power theft detection based on the wide and deep convolutional neural network (CNN). This method can accurately determine the non-periodic and periodic thefts of normal customers and store the general characteristics of the consumed data in one dimension. Ghasemi and Gitizadeh [9] discussed a combined method for detecting unauthorized customers. In this method, based on the probabilistic neural network and Levenberg-Marquardt method, a classification method of customer consumption pattern and a mathematical model are designed in which distributed generation (DG) resources are analyzed on unauthorized consumption. Guerrero et al. [10] proposed a two-module framework for reducing non-technical losses. The first module filters customers based on text mining and a complementary artificial neural network. The second module uses a data mining process including classification and regression tree and self-organization map (SOM). By using these modules, non-technical losses are significantly reduced. Messinis and Hatziargyriou [11] reviewed the recent research papers on the detection of non-technical losses and focused on algorithms, data types and size, features, evaluation metrics, and response times of the NTL detection system. Besides, they provided a comprehensive list of performance metrics and their importance. Maamar and Benahmed [12] provided an overview of machine learning research to detect power theft using smart meter data and studied the challenges of electrical theft detection. Learning models were compared in terms of performance metrics, analytics and simulation environments, and datasets used. Singh et al. [13] employed relative entropy to measure the similarity between possible distributions derived from the reconstructed consumption dataset. When power theft attacks are injected into advanced metering infrastructure (AMI), the energy consumption probability distribution is derived from the consumption history, resulting in greater relative entropy. Nazmul Hasan et al. [14] proposed a hybrid power theft detection system that combines CNN with long short-term memory (LSTM). Since power consumption is time-series data, the CNN-LSTM model was used for classification, and a technique based on local values was employed to calculate missing items. In their study, Li et al. [15] proposed a hybrid model based on CNN and random forest (RF) to detect power theft. In this model, CNN was first used to learn features between different hours of the day and different days extracted from smart meter data. A dropout layer and a back-propagation algorithm were used in this model to reduce over-fitting and update network parameters in the training phase, respectively. The RF was then trained based on the characteristics obtained to determine whether the customer was robbing or not. Konstantinos and Georgios [16] proposed a combined data mining method to identify abnormal customers. In this method, a combination of principal component analysis (PCA) technique and mean shift algorithm were used for different theft scenarios, which detected theft with high efficiency. In their paper, Maamar and Benahmed [17] proposed a hybrid approach based on k-means and deep neural networks to detect power theft anomalies. K-means was used to identify a group of customers with similar power consumption patterns in order to know the different types of normal behavior, and a deep neural network was used to build an anomaly detection model. Chandel and Thakur [18] developed an electrical theft detection methodology based on the detection of abnormal readings of voltage, current, and power factor sent to the center every 15 minutes at regular intervals by smart meters. High detection rate (DR) and low false positive rate (FPR) indicated the high efficiency of the presented method. Razavi et al. [19] provided a feature engineering framework for detecting power theft in smart grades. The framework combined the finite mixture model clustering for customer segmentation and the genetic programming algorithm to identify new features. In this framework, the gradient boosting algorithm was used, which significantly improved the detection rate. Feng et al. [20] presented an algorithm for detecting abnormal power consumption patterns based on local matrix reconstruction (LMR). Five daily load characteristics were used instead of high daily amplitude load curves and PCA technique to calculate weighted regeneration errors. Zhang et al. [21] proposed a feature engineering-based method for detecting abnormal power consumption behavior. First, the main feature set was created by brainstorming. Then, the optimal feature set, which reflected the consumption behavior of customers, was obtained based on the variance and similarity between the selected features. In the abnormal detection stage, a density-based clustering algorithm was used to detect abnormal consumption behaviors.

Ghaedi et al. [22] used a combination of clustering and classification to detect power theft. The improved crow search algorithm was used for optimal customer clustering and the two-stage stacking technique was used to classify electricity consumers.

The effectiveness of any classification technique depends on the features used in the training and testing patterns. Feature selection [23][24] is a technique for selecting the appropriate subset of features that enhances the capabilities and robustness of the models. Indeed, the purpose of feature selection is to improve the performance of models. Over the last few years, many researchers have used ensemble classifiers to increase the efficiency of models. In fact, the methods of ensemble classifiers combine the prediction results of several classifiers to fix the problems related to the base classifiers in order to achieve a better result. In ensemble

classifiers, each classifier gives a different solution to the problem according to its performance and it is expected that by combining these answers, prediction accuracy increases [14]. Two different classifiers may have a variety of errors when patterns that are incorrectly classified are different. The difference in the errors of base classifiers is caused by classifiers covering each other's errors. Important challenges are the proper selection of base classifiers and how to combine them.

In our study, the stacked ensemble technique [25], which is a two-level technique, is used to predict power theft. One of the most important and effective factors in ensemble methods is the correct selection of the base classifiers. To optimally select the base classifiers, the *F-score* and diversity criteria are used together. To improve the prediction of the stacking technique, the performance of the base classifiers is improved by selecting the appropriate features using the CHOA algorithm. The results of the base classifiers are combined based on weighted voting. Each model has a different role in decision-making based on its capabilities. The model used in the second level of stacking is the SLP model, which has generalized properties and is never over-fit. Experiments are performed on the ISSDA dataset [26].

This paper is organized as follows. The cheetah optimization algorithm is described in Section II. Section III reviews the cheetah algorithm for feature selection. Section IV explains the proposed framework for power theft detection. The performance evaluation criteria are explained in Section V. Section VI presents the results and evaluation followed by some concluding points in Section VII.

II. CHEETAH OPTIMIZATION ALGORITHM (CHOA)

Cheetahs are expert hunters built for speed. They have small heads, long legs, and muscular tails to maintain balance. Their hearts and lungs are larger than usual to supply oxygen to the running muscles. Half of their muscle mass is placed around the spine so that they can be as flexible as a spring. This lengthens their strides, which helps them reach their maximum speed [27]. For cheetahs, only three strides are enough to reach a speed from zero to 65 km/h, and it only takes 3 seconds to reach a speed of 110 km/h. Cheetahs need coordinated senses to hunt. Although cheetahs are very fast, they must be close enough to their *victim* before attacking. When they see *victim*, they are completely focused [28]. Teamwork increases the likelihood of success in hunting. When one gets tired, the other continues to chase. Male cheetahs usually hunt in groups [29]. Fig. 1 shows photos of group hunting by cheetahs.

The CHOA algorithm is a population-based meta-heuristic algorithm in which the location and velocity of the $object_i$ ($cheetah_i$ or $victim_i$) in the search space are specified as follows:

$$(location_{object_i} = [location_{object}{}_1^i, location_{object}{}_2^i, \dots, location_{object}{}_d^i]$$
$$velocity_{object_i} = [velocity_{object}{}_1^i, velocity_{object}{}_2^i, \dots, velocity_{object}{}_d^i]$$

where d is the dimension of the search space.

Fig. A1 in Appendix A shows the steps of the CHOA algorithm. All cheetahs choose a direction according to the movement of the *victim*, and the *victim* chooses its direction according to the $cheetah_{leader}$. Based on the current velocity of a cheetah and its distance from the *victim*, a new velocity is calculated for a cheetah. To find the best solution by $cheetah_{leader}$, firstly, all of the *N* cheetahs and *M* victims are initialized at random locations in the search space. Because the cheetahs are not far apart in a group, they are located in a range with a radius $R_{Hunting}$(10 meters).



Fig. 1. Photos of group hunting by cheetahs

At the beginning of the movement (first step), the velocity of all cheetahs and *victim* is considered zero. In the next steps, the velocities and locations are updated. Fig. 2 shows the vectors that determine the movement path of the cheetahs.

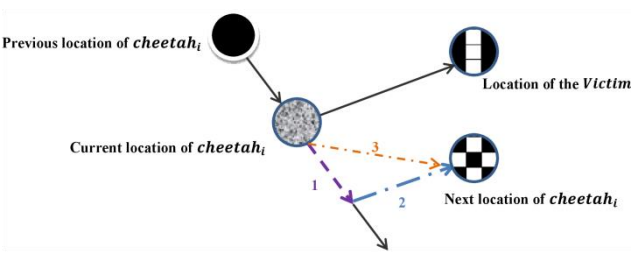


Fig. 2. The direction of the cheetah movement

As shown in Fig. 2, vector 1 represents the coefficient of the previous movement of the $cheetah_i$. Vector 2 represents the coefficient of the vector of the current location to the *victim*. Finally, based on the results of the two vectors, the new velocity of the $cheetah_i$ is determined as vector 3.

For each cheetah, the fitness of the new location is evaluated, and depending on the obtained value, the

$\text{cheetah}_{\text{Leader}}$ may be updated. If the $cheetah_{Leader}$ improves, the *victim* will escape and its velocity and location will be updated.

Fig. 3 also illustrates the movement pattern of the *victim* with respect to the changes in the movement of the $cheetah_{leader}$.

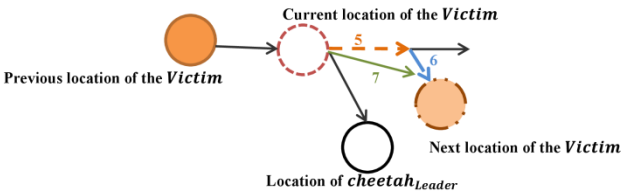


Fig. 3. The direction of the $victim$ movement

As shown in Fig. 3, vector 5 represents the coefficient of the previous movement of the *victim*. Vector 6 represents the coefficient of the vector of the current location to the $cheetah_{Leader}$. Finally, based on the result of the two vectors, the new velocity of the *victim* is determined as vector 7.

To attack, the distance of the cheetahs to the $victim$ must reach a certain threshold. The best distance is between 30 to 100 meters. Therefore, the locations of the victims are randomly adjusted in the mentioned range. One of the important criteria for starting an attack is *Attention*. The $Attention$ of each object (*cheetah* or *victim*) is calculated as follows:

$$Attention_{object_i} = \frac{Fitness(object_i)}{\sum_{i=1}^{N} MAX(Fitness(object_i))} \quad (1)$$

where $Fitness(object_i)$ is the fitness value of $object_i$ and $\sum_{i=1}^{N} MAX(Fitness(object_i))$ represents the highest amount of fitness among objects. In fact, values of the *Attention* are between 0 and 1. To attack, the $cheetah_{Leader}$ must have more *Attention* than *victim*. The cheetah with the highest fitness is known as the $cheetah_{Leader}$. The movement of each cheetah is affected by the movement of the *victim* and is updated as follows:

$$velocity_{cheetah_i} = \omega * velocity_{cheetah_i} + rand1() * \alpha * \left(location_{victim} - location_{cheetah_i}\right) \quad (2)$$

$$New - location_{cheetah_i} = location_{cheetah_i} + velocity_{cheetah_i} \quad (3)$$

where ω is the inertia weight. The weight of inertia controls the effect of past velocities on present velocities. In this study, the weight of inertia decreases linearly. The algorithm usually starts moving with a large amount of the weight inertia, which causes a large search space at the beginning of the algorithm, and this weight decreases over time, which causes the search within a small space at the final steps. In fact, inertia indicates how much the cheetah wants to maintain its current state of motion. A lower amount of inertia results in faster convergence of the algorithm, and increasing the amount of inertia increases the number of sudden movements of the cheetahs. $velocity_{cheetah_i}$ represents the velocity of the $cheetah_i$. α indicates the influence of the location of the $location_{victim}$ on the location of the $cheetah_i$. $location_{victim}$ shows the location of the *victim.* $\text{Location}_{\text{cheetah}_i}$ represents the current location of the cheetah_i. *rand1()* is a random function with a uniform distribution between 0 and 1 that is used to increase random search and maintain the random nature of the algorithm.

During the hunting process, if $cheetah_{Leader}$ improves his fitness, it will not be able to catch the *victim* and the new location of the *victim* will be updated as follows:

$$velocity_{victim} = \gamma * velocity_{victim} + rand2() * \beta * \left(location_{victim} - location_{cheetah_{Leader}}\right) \quad (4)$$

$$New - location_{victim} = location_{victim} + velocity_{victim} \quad (5)$$

where γ is the inertia weight, $location_{victim}$ is the current location of the *victim*,. β is the change percentage in the fitness improvement of $\text{cheetah}_{\text{Leader}}$,. and *rand2()* is a random function with a uniform distribution between 0 and 1. The fact that $cheetah_{Leader}$ is selected based on the distance and attention factors prevents the algorithm from falling into the local optimization. The two most important factors in meta-heuristic algorithms are diversification and intensification. In the CHOA algorithm, with increasing the attention of the *victim*, the cheetah has to search a wider space, and diversification increases. Also, if the attention of the *victim* is low, the cheetah will hunt locally, which means that the intensification will increase.

III. CHEETAH ALGORITHM FOR FEATURE SELECTION

Feature selection is a method of dimension reduction that is used in various fields such as data mining, machine learning, pattern recognition, and so on. In fact, feature selection is a technique of selecting an effective feature subset from an original feature set to increase prediction accuracy. Datasets contain many redundant, irrelevant, and noisy features. Eliminating these features by selecting the important features reduces computational costs so that important information is not lost and learning efficiency is increased. Although many methods have been proposed to solve the problem of selecting important features, most are suffering from the problem of high complexity, high computational cost, and early convergence. Therefore, the use of meta-heuristic algorithms to solve these challenges has received much attention so that today it is one of the most efficient methods and can extract the best subset of features

by maintaining the accuracy of the model and creating more generalizable models [30].

Meta-heuristic search algorithms are inspired by physical, biological, and natural processes, and most of them operate as a population. Unlike classical methods, meta-heuristic search methods operate on a random basis and search space in parallel. These types of methods only use the fitness function to direct the search, but because of their collective intelligence, they can discover the solution.

Various studies have been conducted on feature selection using meta-heuristic algorithms. Pandey et al. [31] used the binary cuckoo binominal search algorithm to select the feature subset. Different versions of the cuckoo search have also been used to select important features [32]–[34]. Naik et al. [35] employed a binary version of the bat algorithm (BBA) with a generalized one-pass neural network to estimate the number of selected features. In some studies [36]–[38], modified versions of BBA have been used to solve various classification problems along with the feature selection strategy. Xu et al. [39] used a combination of a binary firefly algorithm with an opposition-based learning algorithm to select important features. Selvakumar and Muneeswaran [40] used a combination of the firefly algorithm with the Bayesian network and C4.5 with feature selection to detect network intrusion. In one study [41], a modified firefly algorithm was used along with the chaos theory used to select the effective features in medical data. Yan et al. [42] used an improved version of the flower pollination algorithm considering group strategy to select effective features in biomedical data. Zhang et al. [43] improved the krill herd algorithm and used it in high dimensional microarray datasets to extract important features. Pathak et al. [44] used the Levy flight gray wolf optimizer (GWO) to select the appropriate features from original datasets. Al-Tashi et al. [45] used the GWO algorithm to select the best features.

In this paper, the CHOA meta-heuristic algorithm is proposed for feature selection due to its outstanding performance in ensuring diversification and intensification, as well as the use of the dynamic Attention factor for each cheetah.

Each cheetah is represented by a binary string as $F = F_1, F_2, \dots F_n \quad n = 1,2,\dots,2$. $F_i$specifies the consumption for the hour *i* that the consumer sends to the center. Values of the features are considered in the range of 0-1. If the corresponding bit value is more than 0.5, it indicates that the feature is selected; otherwise, it indicates that the feature is not selected.

In fact, each cheetah represents a subset of features that is the candidate solution. The fitness of each cheetah is evaluated using base classifiers and cheetah with the highest fitness selected as the $cheetah_{Leader}$. If cheetah fitness is greater than the $\text{cheetah}_{\text{Leader}}$ fitness, then the location vector for the $cheetah_{Leader}$ is saved and the $\text{cheetah}_{\text{Leader}}$ is updated. Finally, after satisfying the algorithm termination conditions, the optimal features that the $\text{cheetah}_{\text{Leader}}$ has stored for each base classifier are extracted.

According to Algorithm B1 in Appendix B, after loading the data, the parameters related to the cheetahs, including locations and velocities, are set. The initial location of the cheetahs is set within a radius of 10 meters and their initial velocity is set to 0. The initial locations of the *victim* are also determined randomly and their initial velocities are set to 0. Then, according to the number of iterations of the algorithm, the location and velocity of the cheetahs, as well as the *victim*, are updated according to Equs. (2)-(5). At each iteration, the fitness(cost) function of each cheetah is evaluated, and the leader may change during the attack. In fact, the location and velocity of the cheetahs are affected by the movement and velocity of the *victim* .To evaluate the fitness of a cheetah, its location and the data are sent to the cost function of the feature selection. To evaluate the performance of the proposed feature selection algorithm, the results are compared with the actual data label of the dataset and the performance criteria are evaluated.

IV. PROPOSED FRAMEWORK FOR POWER THEFT DETECTION

Fig. C1 in Appendix C shows the steps of the proposed framework. According to Fig. C1, the reference dataset is normalized by the Min-Max method [46] to increase the prediction efficiency.

After data normalization, to make a balance between normal and abnormal samples of the dataset and increase the rate of correct theft detection, abnormal samples have been added to the original dataset according to equations in [2]. The sampling rate and reading of meters have been reduced and done every hour to increase the privacy of customers. For each sample $x = \{x_1, \dots, x_{t \quad t = 1,2,\dots,24}\}$, six types of artificial attacks are generated as follows:

in which the function $h_1(x_t)$ multiplies all samples in a random value and the function $h_2(x_t)$ multiplies each sample in different random values.

$$h_1(x_t) = \alpha x_t \tag{6}$$

$$\alpha = random(0.1,0.8)$$

$$h_2(x_t) = y_t x_t \quad = random(0.1,0.8) \tag{7}$$

$$h_3(x_t) = \begin{cases} 0 & \forall t \in [ts, tf] \\ x_t & else \end{cases} \tag{8}$$

$$h_4(x_t) = \quad y_t mean(x) \tag{9}$$

$$y_t = random(0.1,0.8)$$

$$h_5(x_t) = \quad mean(x) \tag{10}$$

$$h_6(x_t) = x_{24-t} \tag{11}$$

The function $h_3(x_t)$ is a by-pass attack that sends a value of 0 at a specified time interval [ts, tf](ts: t start, tf: t final); otherwise, it sends the actual amount of consumption. ts and

tf are the beginning and ending of the theft period, respectively.

The functions $h_4(x_t)$ and $h_5(x_t)$ focus on sending the average daily consumption. The function $h_6(x_t)$ reverses the order of the readings in a day.

To select the best base classifiers of the staking technique, the *F-score* values of all classifiers are first measured to predict power theft on the dataset. The classifier with the highest *F-score* is selected as the first base classifier. Then, to select the other base classifiers, the triple combinations of the classifier with the highest *F-score* and other classifiers are created. This combination is obtained based on diversity criteria. Diversity criteria include *disagreement*, *Kappa* and *Matthews correlation coefficient* (*MMC*). These criteria are explained in Section V. Each combination with the highest *F-score* is selected as the final combination of the base classifiers.

After identifying base classifiers, the two-level stacking technique is used for training. In the first level of the training, the base classifiers are trained with the same data (56% of the total data) to learn and solve the final problem. According to the ISSDA dataset and to reduce the sampling rate, the number of features is reduced to 24 features. To better train these classifiers, the effective features of each classifier are automatically selected by the CHOA algorithm in this level of training. Features, with values greater than 0.5, are selected. The base classifiers are trained with selected features and their fitness functions are obtained to evaluate. In the second level of training, the meta-model is trained to calculate the weight of each base classifier. These weights determine the degree of importance of the votes of the base classifiers in the final decision. The model used in the second level of training is the single-layer perceptron model, which is characterized by generalizability and is not over-fitted. This model assigns a weight to the output of each base classifier. In fact, the output of the training phase of the meta-model is the weight of each base classifier. After training the base classifiers, the validation data (24% of the total data) and the trained base classifiers are used to generate the training data ($\mathrm{Xtrain_i}$) of the meta-model. Then, the trained meta-model and the weight ($\mathrm{W_i}$) of each base classifier are determined. Finally, the trained model is tested using test data, and the performance of the proposed framework is evaluated.

V. PERFORMANCE EVALUATION CRITERIA

To evaluate the diversity of the classifiers, the average values of all classifier pairs are used as a criterion to compare the diversity of the classifiers. These criteria are obtained based on Table I.

In Table I, X and W are the number of cases in which both classifiers D_i and D_j have correct and incorrect predictions, respectively, Y is the number of cases that classifiers D_i and D_j have negative and positive values, respectively, and Z is the number of cases that classifiers D_i and D_j have positive and negative values, respectively.

Table I

A 2× 2 MATRIX OF THE RELATIONS BETWEEN CLASSIFIERS D_i AND D_j

	D_j correct(1)	D_j wrong(0)
D_i correct(1)	X	Z
D_i wrong(0)	Y	W

Matthews correlation coefficient (MMC):

MMC [47] is one of the metrics related to the correlation measurement between the observed values by the two classifiers and their predicted values. *MMC* values are between -1 and +1. Values close to +1 indicate an ideal prediction and a strong positive correlation between true labels and prediction. The strong correlation indicates that the predicted values are very similar to the actual values. A value of 0 indicates random data prediction.

$$MMC_{ij} = \frac{xw - yz}{\sqrt{(x+z)(x+y)(y+w)(z+w)}} \tag{12}$$

A lower value of this criterion shows more diversity between the two classifiers.

Disagreement measure:

Disagreement [41][48] represents the ratio between the number of observations that one classifier is correct and the other is incorrect according to the total number of observations. A greater value of this criterion shows more diversity between the two classifiers.

$$D_{ij} = \frac{z+y}{x+y+z+w} \tag{13}$$

Kappa coefficient:

Kappa is one of the important criteria to evaluate the diversity between the two observers from the perspective of agreement [49]. *Kappa*'s main purpose is to evaluate the degree of agreement or disagreement of two observers from a similar event. In fact, *Kappa* is based on the idea of measuring agreement between true and predicted labels and has been proposed to remove chance agreement from overall *accuracy*.

$$K_{ij} = \frac{\Theta_1 - \Theta_2}{1 - \Theta_2} \tag{14}$$

where Θ_1 is overall accuracy (observed agreement) and Θ_2 is the chance agreement (agreement expected only by chance).

$$\Theta_1 = \frac{TP + TN}{N} \quad (15)$$

$$\Theta_2 = \frac{(TP + FN)(TP + FP) + (TN + FP)(TN + FN)}{N^2} \quad (16)$$

In calculating the kappa coefficient, in addition to the data that are observed correctly, the data that are observed incorrectly are also included. Then, it is a good measure for comparing the results of two classifiers. The *kappa* coefficient can be between -1 and +1. A lower value shows lower agreement and higher diversity.

To evaluate the efficiency of the proposed framework, several performance criteria have been used, i.e., *accuracy*, *recall*, *precision*, *F-score*, specificity, and receiver operating characteristic (*ROC*) curve. These criteria are based on the confusion matrix presented in Table II.

Table II

CONFUSION MATRIX

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Abbreviations in the confusion matrix are true positive (*TP*), true negative (*TN*), false positive (*FP*), and false negative (*FN*). In general, $N = TP + TN + FN + FN$.

Accuracy:

Accuracy is the correct detection rate of the classifier in the total of the classes:

$$Accuracy = \frac{(TP + TN)}{N} \quad (17)$$

Recall (sensitivity):

Recall represents the ratio of the number of positive data in a particular class that is correctly classified to the number of data that should be classified in that particular class.

$$Recall = \frac{TP}{TP + FN} \quad (18)$$

Precision:

This criterion expresses the ratio of the number of correct predictions made for samples of a particular class to the total number of predictions for samples of the same class.

$$Precision = \frac{TP}{TP + FP} \quad (19)$$

F-score:

This criterion combines precision and recall parameters to better evaluate the performance of a classification model. This criterion provides a more accurate description of how the classification model works on all existing classes in the data.

$$F - score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (20)$$

Specificity:

Specificity indicates the ratio of the number of negative data in a particular class that is correctly classified to the number of data that should be classified in that particular class.

$$Specificity = \frac{TN}{TN + FP} \quad (21)$$

Receiver operating characteristic (ROC) curve:

An excellent prediction is a prediction in which the values of both sensitivity and specificity criteria are 1. These two criteria are always in competition with each other. Increasing one is accompanied by decreasing the other. This situation led to the production of the *ROC* curve, which expresses the relationship between the two criteria of sensitivity and specificity. The vertical axis of this curve indicates *sensitivity* and the horizontal axis represents *1-specificity*. The area under this curve (*AUC*) is used as a measure to evaluate the performance of the classifier. Ideally, the area under the *ROC* curve is 1.

VI. RESULTS AND EVALUATION

In this section, the results of the proposed framework are evaluated on the ISSDA dataset. In all experiments, the average of different runs was used to measure the evaluation criteria. The ISSDA dataset contains consumption data of 5,000 Irish home and business customers from 2009 to 2010, published in January 2012. For each customer, consumption information is stored for 535 days so that each day contains 24 features. Table III shows the information stored in this dataset.

Table III

INFORMATION OF ISSDA DATASET

Meter ID	Consumption time (day-hour)	Consumption amount (Kwh)

Table IV shows the information of a meter, including the meter ID, consumption value in each hour (kWh), and consumption date that are sent to the center.

Table IV

CONSUMPTION INFORMATION OF A METER

Date	Hour1(kWh)	…	Hour24(kWh)	Meter ID
2009/29/8	0.094	…	0.149	1184

For example, Hour1(kWh) represents the amount of consumption electricity at 01:00 a.m.

Base classifiers are selected among the classifiers of SVM, neural network (NN), NB, AdaBoost [50], RF, and KNN. According to Table V, the *F-score* criterion of these classifiers was obtained to predict power theft. The final base classifiers of the stacking technique are considered to be three classifiers. According to Table V, the SVM classifier has the highest *F-score*. To select base classifiers, the triple combinations of the SVM classifier with the highest *F-score* and other classifiers are created. The values of diversity

criteria are calculated based on the average values of each pair of classifiers. As shown in Table VI, according to the *Kappa* and *MMC* criteria, the SVM-NB-AdaBoost combination has the best diversity.

Table V

F-SCORE CRITERION OF THE BASE CLASSIFIERS (NO-FEATURE SELECTION)

Classifier / Criterion	SVM	NN	NB	AdaBoost	RF	KNN
F-score	0.897	0.893	0.864	0.877	0.862	0.871

Also according to the disagreement criterion, this combination has the best diversity after the SVM-NB-RF combination. Given that the SVM-NB-AdaBoost combination has the highest diversity, this combination is used for base classifiers.

According to Table V, the SVM, NN, and AdaBoost classifiers have the highest *F-score*. If their combination is used as the base classifier in the proposed framework, they are less efficient than when both the *F-score* and diversity criteria are used.

Table VI

F-SCORE AND DIVERSITY VALUES OF THE SVM AND OTHER CLASSIFIERS

Subset classifiers	AVG Kappa ↓	AVG disagreement ↑	AVG MMC ↓	F-score ↑
SVM-NB-KNN	0.8941	0.0085	0.8985	0.8918
SVM-NB- RF	0.8497	0.0126	0.8556	0.9245
SVM-NN-KNN	0.9026	0.0074	0.9270	0.8732
SVM-NN-RF	0.8736	0.0101	0.8796	0.9113
SVM-KNN-RF	0.8720	0.0105	0.8776	0.9137
SVM-RF-AdaBoost	0.9057	0.0074	0.9095	0.8862
SVM-NN-AdaBoost	0.8994	0.0074	0.9008	0.8720
SVM-KNN-AdaBoost	0.8923	0.0085	0.8953	0.9144
SVM-NB-AdaBoost	0.8474	0.0126	0.8525	0.9624
SVM-NB-NN	0.8911	0.0085	0.8953	0.9045

As is seen in Table VI, the SVM-NB-AdaBoost combination, which has better diversity, also has the highest *F-score*. After selecting the SVM, NB, and AdaBoost classifiers as the base classifiers, out of 3745 data in the dataset, 80% were randomly selected for training and 20% for testing. Out of the total training data, 80% were used for training the base classifiers and 20% as validation data to train and produce data for Meta classifier. The model used in the second level of the stacking technique is SLP, which has the generalization property and is never over-fit.

After training the second level classifier, weights of the base classifiers are obtained. The results of the first-level classifiers are combined based on the weighted voting.

A. *Experiment 1*

In this experiment, to increase the prediction performance of the proposed framework, the CHOA algorithm was used to select the features of the SVM, NB, and AdaBoost classifiers.

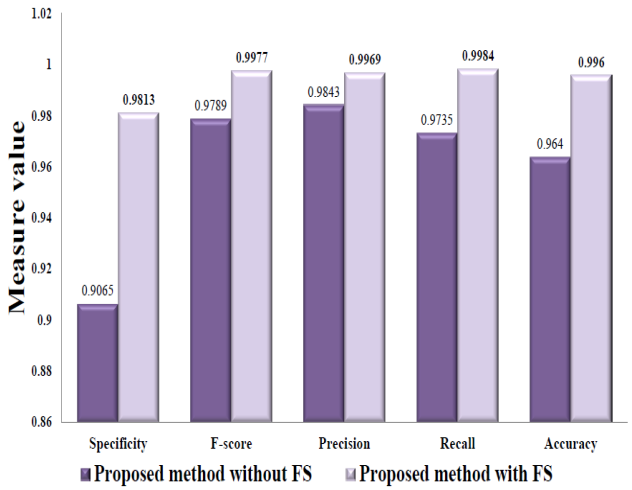


Fig. 4. Values of the *accuracy*, *recall*, *precision*, *F - score*, and *specificity* criteria for both conditions of feature and no-feature selection

The values of *accuracy*, *recall*, *precision*, *F-score*, and *specificity* criteria were evaluated for feature selection/no feature selection conditions.

Fig. 4 presents the values of the mentioned criteria for both conditions of feature selection and no-feature selection. According to the results in Fig. 4, it can be seen that the values of evaluation criteria are higher when the feature selection strategy is used. For example, the accuracy criterion value is 0.996 for the feature selection method and 0.964 for the strategy of no-feature selection.

In fact, with the feature selection strategy, abnormal power customers were identified with fewer features and higher efficiency. The selected features of the classifiers are:

AdaBoost: Hour8, Hour9, Hour10, Hour11, Hour12, Hour13, Hour14, Hour15, Hour16, Hour20, Hour21, Hour22, Hour23, Hour24.

NB: Hour7, Hour8, Hour9, Hour10, Hour11, Hour12, Hour13, Hour14, Hour15, Hour19, Hour20, Hour21, Hour22, Hour23.

SVM: Hour6, Hour7, Hour8, Hour9, Hour10, Hour11, Hour12, Hour13, Hour14, Hour15, Hour16, Hour17, Hour20, Hour21, Hour22, Hour23, Hour24.

B. *Experiment 2*

In this experiment, the *ROC* curves of the proposed framework were illustrated for situations without or with feature selection.

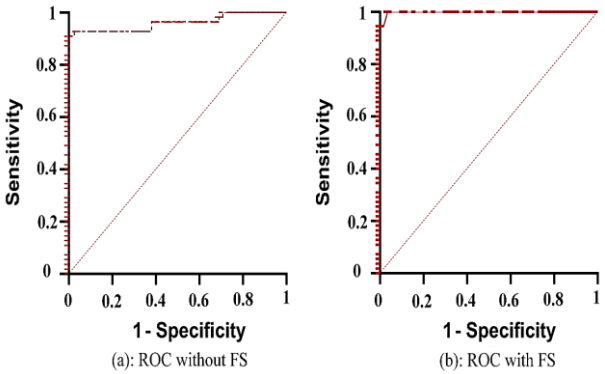


Fig. 5. The *ROC* curves for both conditions of the feature selection and without feature selection

According to Fig. 5, it can be seen that the proposed framework for the situation where the feature selection algorithm is used to detect power theft shows a higher efficiency. According to the *ROC* curves in Figs. 5(a) and 5(b), the values of the *AUC* parameter for situations without feature selection and with feature selection are 0.9632 and 0.9982, respectively. The obtained *AUC* value for the feature selection situation indicates the high performance of the proposed framework in detecting abnormal power customers.

C. Experiment 3

In this experiment, the results of the proposed framework were compared with the results reported in [22] and an average of 20 different runs used to evaluate the results and measure the criteria. In each independent run, the maximum iteraion is set to 100. The attacks of the research [2] have been used to evaluate the results. For all the methods in Table VII, plan 1 (all attacks) and scenario of 80% of the data for training and 20% for testing were used. Table VII shows the values of the *accuracy* criterion for this experiment. According to Table VII, it can be seen that the *accuracy* value of the proposed framework is 0.996, which indicates the higher efficiency of the proposed framework to detect power theft.

Table VII

COMPARISON OF THE ACCURACY CRITERION OF THE PROPOSED FRAMEWORK AND RESEARCHES [22]

Research \ Criterion	*Accuracy*
ICSA-Stacking [22]	0.9835
Proposed framework	0.996

D. Experiment 4

In this experiment, the results of the proposed framework were compared with the results of the research [15].

Table VIII shows the values of the *recall*, *precision*, and *F-score* criteria for both frameworks.

Table VIII

COMPARISON OF RECALL, PRECISION, AND F – SCORE CRITERIA OF THE PROPOSED FRAMEWORK AND RESEARCH [15]

Criterion \ Method	CNN-RF	Proposed Framework
Recall	0.97	0.9984
Precision	0.97	0.9969
F-score	0.97	0.9977

According to Table VIII, it can be seen that the values of $recall$, $precision$, and F-score criteria for the proposed framework are higher than the results reported in [15]. With these results, we find that the proposed method identifies unauthorized electricity customers with higher efficiency.

E. Experiment 5

In this experiment, the prediction results of the proposed framework were compared with different feature selection algorithms. PSO, crow search algorithm (CSA) [51], and improved CSA (ICSA) [22] feature selection algorithms were used to compare the results.

According to Table IX, it can be seen that the proposed framework with the CHOA feature selection algorithm is more efficient in detecting abnormal consumers.

Table IX

COMPARISON OF DIFFERENT FEATURE SELECTION ALGORITHMS IN COMBINATION WITH THE PROPOSED FRAMEWORK

Criterion \ Method	PSO	CSA	ICSA	CHOA
Accuracy	0.9017	0.922	0.942	0.996

According to Fig. 6, CHOA can converge the fastest due to its low complexity, and in the ISSDA dataset, none exceeds 10 iterations. This illustrates why it is applied for the feature selection algorithm.

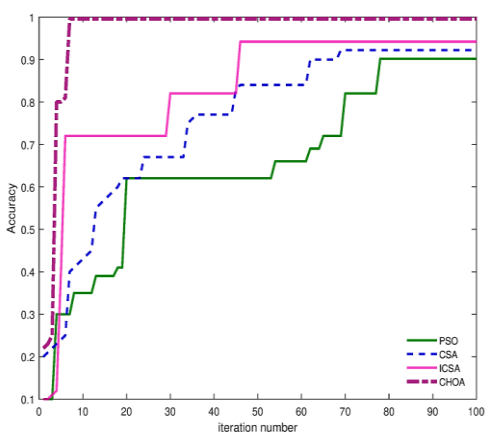


Fig. 6.The converged curves for different feature selection algorithms

Other algorithms converge very slowly, requiring at least 80 iterations to converge. The proposed algorithm also converges very fast, and basically, 10 iterations are sufficient to get a good solution and detect power theft. Figs. 6 illustrates the converged curves for PSO, CSA, ICSA, and CHOA feature selection algorithms.

VII. CONCLUSION

In this research, a two-level stacking technique was used to detect power theft. The base classifiers of the first level of this technique were selected using *F-score* and diversity criteria. The *disagreement*, *Kappa* and *Matthews correlation coefficient* criteria were used to evaluate the diversity of the classifiers. Among the SVM, NN, NB, AdaBoost, RF, and KNN classifiers, the base classifiers of SVM, NB, and AdaBoost were selected. To increase the correct detection rate of this technique, the CHOA algorithm was used to select the features of the base classifiers. The results of the proposed framework were compared with the results of various studies, which showed that the efficiency of the proposed framework was higher than other studies.

Appendix A:

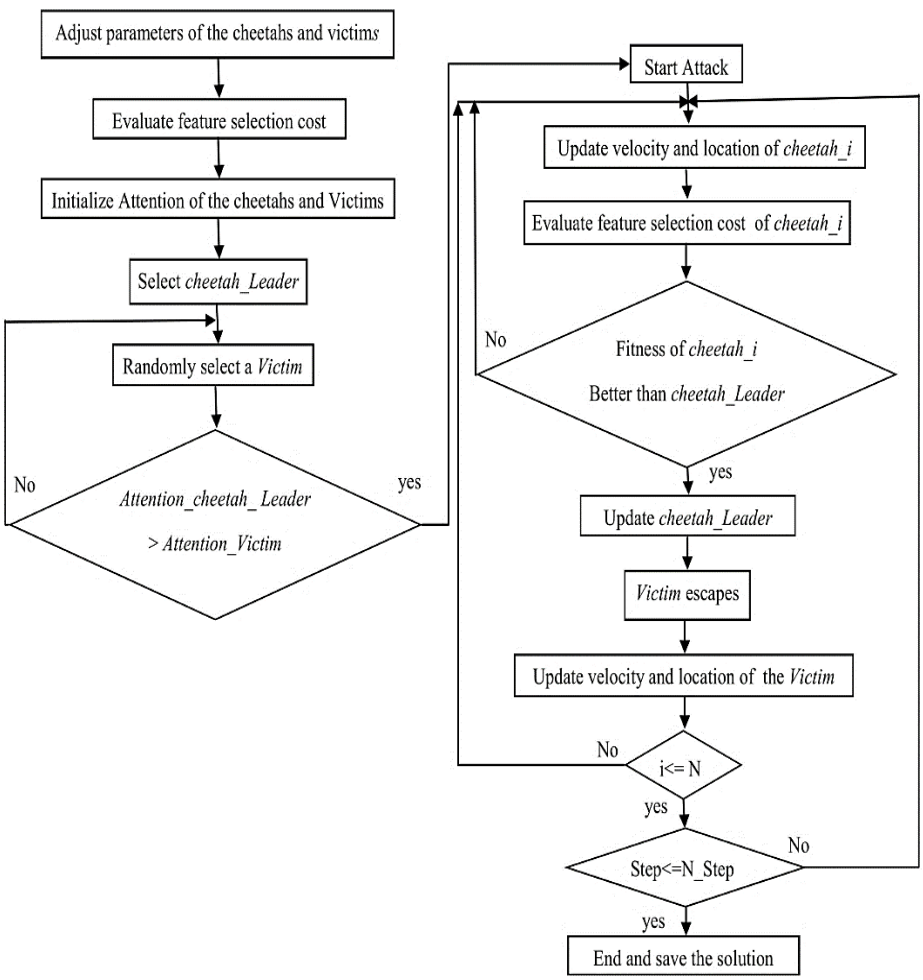


Fig. A1. Steps of the CHOA algorithm

Appendix B:

- **Load data**
- **Adjust parameters of the cheetahs and victims**

Randomly initialize the location of N cheetahs in the search space with radius $R_{hunting}$(10 meters);
Randomly initialize the location of M victims in the distance of 30 to 100 meters from the area of cheetahs;
Adjust the velocity of cheetahs and victims to 0;

- **Evaluate the fitness of each cheetah (feature selection cost)** ;
- **Calculate the *Attention* of each cheetah and *victim***;
- **Select cheetah with highest fitness as** $cheetah_{Leader}$;
- **Randomly select one of the victims as *victim***;

if $Attention_{cheetah_{Leader}} > Attention_{victim}$ **then**
  Attack is start;
end
else
  select another *victim*;
end
Set step=1;
while $step < Nsteps$ **do**
  • On the way to *victim*, the velocity and location of each cheetah are updated as follows:
  for i=*1 to* N **do**
    $velocity_{cheetah_i} = \omega \times velocity_{cheetah_i} + rand1() \times \alpha \times (location_{victim} - location_{cheetah_i})$;
    $New - location_{cheetah_i} = location_{cheetah_i} + velocity_{cheetah_i}$;
    • Evaluate the fitness of $cheetah_i$ (feature selection cost);
    if *fitness of* $cheetah_i$ *is better than* $cheetah_{Leader}$ **then**
      •Update the $cheetah_{Leader}$;
      • *victim* escapes and its velocity and location are updated as follows:
      $velocity_{victim} = \gamma \times velocity_{victim} + rand2() \times \beta \times (location_{victim} - location_{cheetah_{Leader}})$;
      $New - location_{victim} = location_{victim} + velocity_{victim}$;
    end
  end
end

Algorithm B1. The pseudo-code of the feature selection using the CHOA algorithm

Appendix C:

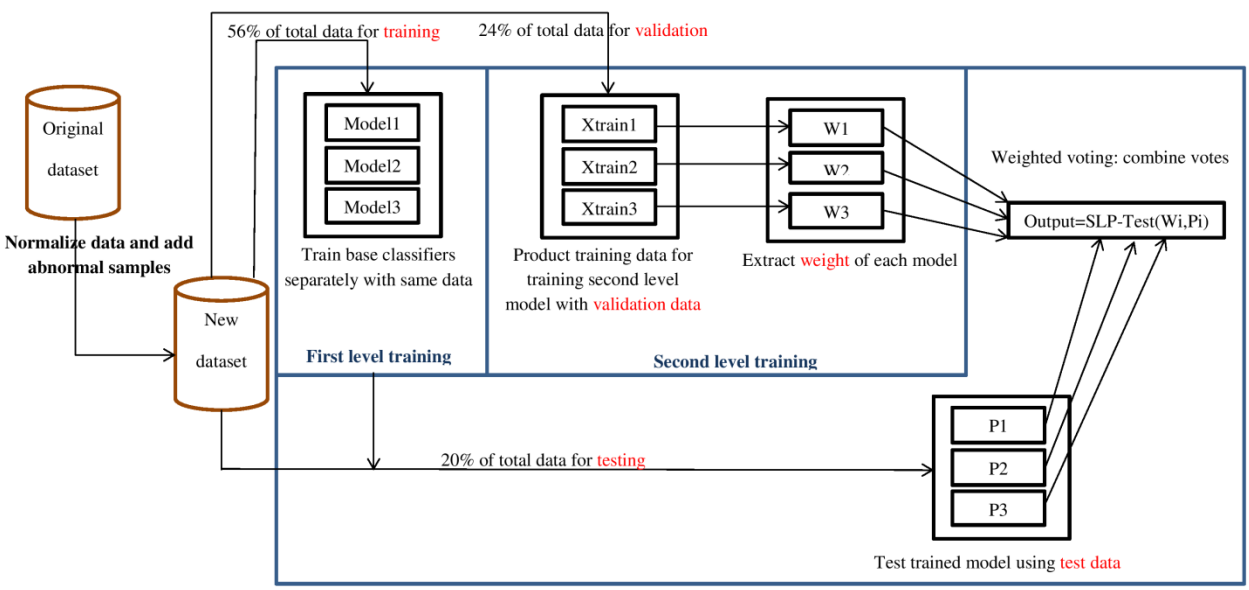


Fig. C1. flowchart of the proposed framework

Mode1= train(FS-AdaBoost(datatrain, dtrain))
Model2= train(FS-NB(datatrain, dtrain))
Model3= train(FS-SVM(datatrain, dtrain))
Xtrain1= Predict(Model1, datavalid)
Xtrain2= Predict(Model2, datavalid)
Xtrain3= Predict(Model3, datavalid)
W1 = Train-SLP(Xtrain1, dvalid)
W2 = Train-SLP(Xtrain2, dvalid)
W3 = Train-SLP(Xtrain3, dvalid)
P1= Predict(Model1, datatest)
P2= Predict(Model2, datatest)
P3= Predict(Model3, datatest)

FS:feature selection
dtatrain: train data for base classifier
dtrain: label of the train data
datavalid: validation data to train second level model
dvalid: label of the validation data
datatest: the test data

REFERENCES

[1] L. A. Passos Júnior *et al.*, "Unsupervised non-technical losses identification through optimum-path forest," *Electric Power Systems Research*, vol. 140, pp. 413–423, 2016, doi: 10.1016/j.epsr.2016.05.036.

[2] P. Jokar, N. Arianpoo, and V. C. M. Leung, "Electricity theft detection in AMI using customers' consumption patterns," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 216–226, 2016, doi: 10.1109/TSG.2015.2425222.

[3] T. Ahmad, H. Chen, J. Wang, and Y. Guo, "Review of various modeling techniques for the detection of electricity theft in smart grid environment," *Renewable and Sustainable Energy Reviews*, vol. 82, no. November 2016, pp. 2916–2933, 2018, doi: 10.1016/j.rser.2017.10.040.

[4] C. Cortes and V. Vapnik, "Support-vector networks," *Chem. Biol. Drug Des.*, vol. 297, pp. 273–297, Jan. 2009, doi: 10.1007/%2FBF00994018.

[5] D. D. Lewis, "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval," 1998.

[6] K. Zheng, Y. Wang, Q. Chen, and Y. Li, "Electricity theft detecting based on density-clustering method," *2017 IEEE Innovative Smart Grid Technologies - Asia: Smart Grid for Smart Community, ISGT-Asia 2017*, pp. 1–6, 2018, doi: 10.1109/ISGT-Asia.2017.8378347.

[7] F. Xiao and Q. Ai, "Electricity theft detection in smart grid using random matrix theory," *IET Generation, Transmission and Distribution*, vol. 12, no. 2, pp. 371–378, 2018, doi: 10.1049/iet-gtd.2017.0898.

[8] Z. Zheng, Y. Yang, X. Niu, H. N. Dai, and Y. Zhou, "Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, 2018, doi: 10.1109/TII.2017.2785963.

[9] A. A. Ghasemi and M. Gitizadeh, "Detection of illegal consumers using pattern classification approach combined with Levenberg-Marquardt method in smart grid," *International Journal of Electrical Power and Energy Systems*, vol. 99, no. December 2017, pp. 363–375, 2018, doi: 10.1016/j.ijepes.2018.01.036.

[10] J. I. Guerrero, I. Monedero, F. Biscarri, J. Biscarri, R. Millan, and C. Leon, "Non-Technical Losses Reduction by Improving the Inspections Accuracy in a Power Utility," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1209–1218, 2018, doi: 10.1109/TPWRS.2017.2721435.

[11] G. M. Messinis and N. D. Hatziargyriou, "Review of non-technical loss detection methods," *Electric Power Systems Research*, vol. 158, pp. 250–266, 2018, doi: 10.1016/j.epsr.2018.01.005.

[12] A. Maamar and K. Benahmed, "Machine learning techniques for energy theft detection in AMI," *ACM International Conference Proceeding Series*, pp. 57–62, 2018, doi: 10.1145/3178461.3178484.

[13] S. K. Singh, R. Bose, and A. Joshi, "Energy theft detection in advanced metering infrastructure," *IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings*, vol. 2018-Janua, pp. 529–534, 2018, doi: 10.1109/WF-IoT.2018.8355148.

[14] M. Nazmul Hasan, R. N. Toma, A. Al Nahid, M. M. Manjurul Islam, and J. M. Kim, "Electricity theft detection in smart grid systems: A CNN-LSTM based approach," *Energies*, vol. 12, no. 17, pp. 1–18, 2019, doi: 10.3390/en12173310.

[15] S. Li, Y. Han, X. Yao, S. Yingchen, J. Wang, and Q. Zhao, "Electricity Theft Detection in Power Grids with Deep Learning and Random Forests," *Journal of Electrical and Computer Engineering*, vol. 2019, 2019, doi: 10.1155/2019/4136874.

[16] B. Konstantinos and S. Georgios, "Efficient Power Theft Detection for Residential Consumers Using Mean Shift Data Mining Knowledge Discovery Process," *International Journal of Artificial Intelligence & Applications*, vol. 10, no. 01, pp. 69–85, 2019, doi: 10.5121/ijaia.2019.10106.

[17] A. Maamar and K. Benahmed, "A Hybrid Model for Anomalies Detection in AMI System Combining K-means Clustering and Deep Neural Network," *Computers, Materials and Continua*, vol. 60, no. 1, pp. 15–39, 2019, doi: 10.32604/cmc.2019.06497.

[18] P. Chandel and T. Thakur, "A novel rule based technique to detect electricity theft in India," *Advances in Science, Technology and Engineering Systems*, vol. 4, no. 2, pp. 413–421, 2019, doi: 10.25046/aj040251.

[19] R. Razavi, A. Gharipour, M. Fleury, and I. J. Akpan, "A practical feature-engineering framework for electricity theft detection in smart grids," *Applied Energy*, vol. 238, no. December 2018, pp. 481–494, 2019, doi: 10.1016/j.apenergy.2019.01.076.

[20] Z. Feng, J. Huang, W. H. Tang, and M. Shahidehpour, "Data mining for abnormal power consumption pattern detection based on local matrix reconstruction," *International Journal of Electrical Power and Energy Systems*, vol. 123, no. February, p. 106315, 2020, doi:

10.1016/j.ijepes.2020.106315.

[21] W. Zhang, X. Dong, H. Li, J. Xu, and D. Wang, "Unsupervised Detection of Abnormal Electricity Consumption Behavior Based on Feature Engineering," *IEEE Access*, vol. 8, pp. 55483–55500, 2020, doi: 10.1109/ACCESS.2020.2980079.

[22] H. Ghaedi, S. R. Kamel Tabbakh Farizaniv, and R. Ghaemi, "Improving power theft detection using efficient clustering and ensemble classification," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 5, pp. 3704–3717, 2021, doi: 10.11591/ijece.v11i5.pp3704-3717.

[23] H. Liu and H. Motoda, *Feature Extraction, Construction and Selection: A Data Mining Perspective*. USA: Kluwer Academic Publishers, 1998.

[24] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005, doi: 10.1109/TKDE.2005.66.

[25] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992, doi: https://doi.org/10.1016/S0893-6080(05)80023-1.

[26] "Irish Social Science Data Archive," 2012. https://www.ucd.ie/issda/data/commissionforenergyregulationcer/.

[27] J. W. Wilson *et al.*, "Cheetahs, Acinonyx jubatus, balance turn capacity with pace when chasing prey," *Biology letters*, vol. 9, no. 5, p. 20130620, Sep. 2013, doi: 10.1098/rsbl.2013.0620.

[28] L. K. Van der Weyde, T. Y. Hubel, J. Horgan, J. Shotton, R. McKenna, and A. M. Wilson, "Movement patterns of cheetahs (Acinonyx jubatus) in farmlands in Botswana," *Biology Open*, vol. 6, no. 1, pp. 118–124, Dec. 2016, doi: 10.1242/bio.021055.

[29] M. S. Farhadinia *et al.*, "Feeding ecology of the Asiatic cheetah Acinonyx jubatus venaticus in low prey habitats in northeastern Iran: Implications for effective conservation," *Journal of Arid Environments*, vol. 87, pp. 206–211, 2012, doi: https://doi.org/10.1016/j.jaridenv.2012.05.002.

[30] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019)," *IEEE Access*, vol. 9, pp. 26766–26791, 2021, doi: 10.1109/ACCESS.2021.3056407.

[31] A. Pandey, D. Rajpoot, and M. Saraswat, "Feature selection method based on hybrid data transformation and binary binomial cuckoo search," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, Feb. 2020, doi: 10.1007/s12652-019-01330-1.

[32] S. Sarvari, N. F. M. Sani, Z. M. Hanapi, and M. T. Abdullah, "An Efficient Anomaly Intrusion Detection Method With Feature Selection and Evolutionary Neural Network," *IEEE Access*, vol. 8, pp. 70651–70663, 2020, doi: 10.1109/ACCESS.2020.2986217.

[33] R. Shetty, S. Alangar, and P. Pai, "An efficient online sequential extreme learning machine model based on feature selection and parameter optimization using cuckoo search algorithm for multi-step wind speed forecasting," *Soft Computing*, vol. 25, Jan. 2021, doi: 10.1007/s00500-020-05222-x.

[34] S. Marso and M. El Merouani, "Predicting financial distress using hybrid feedforward neural network with cuckoo search algorithm," 2020.

[35] A. Naik, V. Kuppili, and D. Edla, "Efficient feature selection using one-pass generalized classifier neural network and binary bat algorithm with a novel fitness function," *Soft Computing*, vol. 24, Mar. 2020, doi: 10.1007/s00500-019-04218-6.

[36] T. Niu, J. Wang, K. Zhang, and P. Du, "Multi-step-ahead wind speed forecasting based on optimal feature selection and a modified bat algorithm with the cognition strategy," *Renewable Energy*, vol. 118, pp. 213–229, 2018, doi: https://doi.org/10.1016/j.renene.2017.10.075.

[37] S. Jeyasingh and M. Veluchamy, "Modified Bat Algorithm for Feature Selection with the Wisconsin Diagnosis Breast Cancer (WDBC) Dataset," *Asian Pacific journal of cancer prevention : APJCP*, vol. 18, pp. 1257–1264, May 2017, doi: 10.22034/APJCP.2017.18.5.1257.

[38] S. Fei, "Fault diagnosis of bearing based on relevance vector machine classifier with improved binary bat algorithm for feature selection and parameter optimization," *Advances in Mechanical Engineering*, vol. 9, p. 168781401668529, Jan. 2017, doi: 10.1177/1687814016685294.

[39] H. Xu, S. yu, J. Chen, and X. Zuo, "An Improved Firefly Algorithm for Feature Selection in Classification," *Wireless Personal Communications*, vol. 102, pp. 1–12, Oct. 2018, doi: 10.1007/s11277-018-5309-1.

[40] S. B and M. K, "Firefly algorithm based Feature Selection for Network Intrusion Detection," *Computers & Security*, vol. 81, Nov. 2018, doi: 10.1016/j.cose.2018.11.005.

[41] S. Dash, R. Thulasiram, and P. Thulasiraman, "Modified Firefly Algorithm With Chaos Theory for Feature Selection: A Predictive Model for Medical Data," *International Journal of Swarm Intelligence Research*, vol. 10, pp. 1–20, Apr. 2019, doi: 10.4018/IJSIR.2019040101.

[42] C. Yan, J. Ma, H. Luo, G. Zhang, and J. Luo, "A Novel Feature Selection Method for High-Dimensional Biomedical Data Based on an Improved Binary Clonal Flower Pollination Algorithm," *Human Heredity*, vol. 84, pp. 1–13, Sep. 2019, doi: 10.1159/000501652.

[43] G. Zhang, J. Hou, J. Wang, C. Yan, and J. Luo, "Feature Selection for Microarray Data Classification Using Hybrid Information Gain and a Modified Binary Krill Herd Algorithm," *Interdisciplinary Sciences: Computational Life Sciences*, vol. 12, May 2020, doi: 10.1007/s12539-020-00372-w.

[44] Y. Pathak, K. V Arya, and S. Tiwari, "Feature selection for image steganalysis using levy flight-based grey wolf optimization," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 1473–1494, 2019, doi: 10.1007/s11042-018-6155-6.

[45] Q. Al-Tashi, H. Rais, and S. Jadid Abdulkadir, *Feature Selection Method Based on Grey Wolf Optimization for Coronary Artery Disease Classification*. 2018.

[46] J. Han, M. Kamber, and J. Pei, "3 - Data Preprocessing," in *The Morgan Kaufmann Series in Data Management Systems*, J. Han, M. Kamber, and J. B. T.-D. M. (Third E. Pei, Eds. Boston: Morgan Kaufmann, 2012, pp. 83–124.

[47] L. I. Kuncheva and C. J. Whitaker, "Measures of

Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003, doi: 10.1023/A:1022859003006.

[48] D. B. Skalak, "The Sources of Increased Accuracy for Two Proposed Boosting Algorithms," 1996.

[49] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012, [Online]. Available: http://europepmc.org/abstract/MED/23092060.

[50] Y. Freund, "Boosting a Weak Learning Algorithm by Majority," *Information and Computation*, vol. 121, no. 2, pp. 256–285, 1995, doi: https://doi.org/10.1006/inco.1995.1136.

[51] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Computers and Structures*, vol. 169, pp. 1–12, 2016, doi: 10.1016/j.compstruc.2016.03.001.



Hassan Ghaedi received the M.S. degrees in software engineering from islamic azad University, Arak branch, iran. He is currently pursuing the Ph.D. degree in the Department of Computer Engineering, islamic azad University, Neyshabur branch, iran. His research interests include data mining (classification and clustering).



Seyed Reza Kamel Tabbakh is with the Department of Software Engineering, Faculty of Engineering, Islamic Azad University Mashhad branch, Mashhad, Iran. He received his PhD in communication and network engineering from University Putra Malaysia (UPM) in 2011. He received his BSc and MSc in software engineering from Islamic Azad University, Mashhad branch and Islamic Azad University, South Tehran branch, Iran respectively. His research interests include IPv6 networks, routing and security.



RezaGhaemi received his Ph.D. degree in Computer Engineering and Artificial Intelligence from National University Putra Malaysia in 2011. He is currently an Associate Professor at Islamic Azad University, Quchan Branch. His research interests include Machine Learning, Data Mining and Soft Computing.