



Enhancing the performance of smart electrical grids using data mining and fuzzy inference engine

Rana Mohamed El-Balka¹ · Ahmed I. Saleh¹ · Ahmed A. Abdullah² · Noha Sakr¹

Received: 3 September 2021 / Revised: 13 January 2022 / Accepted: 27 March 2022 /

Published online: 17 April 2022

© The Author(s) 2022, corrected publication 2022

Abstract

This paper is about enhancing the smart grid by proposing a new hybrid feature-selection method called feature selection-based ranking (FSBR). In general, feature selection is to exclude non-promising features out from the collected data at Fog. This could be achieved using filter methods, wrapper methods, or a hybrid. Our proposed method consists of two phases: filter and wrapper phases. In the filter phase, the whole data go through different ranking techniques (i.e., relative weight ranking, effectiveness ranking, and information gain ranking) The results of these ranks are sent to a fuzzy inference engine to generate the final ranks. In the wrapper phase, data is being selected based on the final ranks and passed on three different classifiers (i.e., Naive Bayes, Support Vector Machine, and neural network) to select the best set of the features based on the performance of the classifiers. This process can enhance the smart grid by reducing the amount of data being sent to the cloud, decreasing computation time, and decreasing data complexity. Thus, the FSBR methodology enables the user load forecasting (ULF) to take a fast decision, the fast reaction in short-term load forecasting, and to provide a high prediction accuracy. The authors explain the suggested approach via numerical examples. Two datasets are used in the applied experiments. The first dataset reported that the proposed method was compared with six other methods, and the proposed method was

✉ Rana Mohamed El-Balka
rana.m.elbalka@gmail.com

Ahmed I. Saleh
aisaleh@yahoo.com

Ahmed A. Abdullah
ahmed.abdelaleem@deltaniv.edu.eg

Noha Sakr
Nohasakr83@gmail.com

¹ Computer Engineering and Systems Department, Faculty of Engineering, Mansoura University, Mansoura, Egypt

² Communication and Computer Engineering, Faculty of Engineering, Delta University for Science and Technology, Belqas, Egypt

represented the best accuracy of 91%. The second data set, the generalization data set, reported 90% accuracy of the proposed method compared to fourteen different methods.

Keywords Cloud computing · Data mining · Features selection (FS) · Fog computing · Internet of things (IoT) · Load forecasting · Smart grids

1 Introduction

A smart grid has become a global trend due to its usage mechanism of information and communication technologies [14]. Smart grids had a simple architecture of two levels in which data are sent from devices directly to the cloud to be processed and the results are sent back. This could not be possible without the Internet of Things (IoT) which is the infrastructure of the whole architecture [11, 32]. IoT made it possible for such systems to be maintainable, secure, flexible, and interactive [36]. Unfortunately, this two-level architecture had many problems such as latency as the whole data is sent to the cloud directly and no control over location as all devices at different locations end to pour in the same cloud [11]. To overcome the previous problems, a new architecture of three levels came to the light. This new level is called fog and it is the second level or the intermediate medium between devices and the cloud.

Fog is responsible to receive data from devices, apply data pre-processing, remove unnecessary (or noisy) data, and apply computations on data based on rules received from the cloud. These rules are responsible to represent the data in an appropriate form; so it can be stored in a local server inside the fog to provide a quick response for real-time applications [51]. Periodically, data is being sent from the local server inside the fog to a larger server managed by the cloud called Cloud Computing Data Center (C2DC). After that, then the data is removed from the local server so it is ready to receive a bunch of new data from devices. Fog succeeded to reduce the load on the cloud and the amount of data being sent to it [12, 33, 51].

Smart grid made it easy and possible to ensure stability in the electrical network by monitoring the relationship between supplies and demands of all users at once and tracking the users' behaviors for different hours, days, or maybe any sudden events. So that, the user is an essential part of the network to maintain its stability and control the required energy size from the stations [19, 52]. However, this shows up as a problem as many of these events or data is unnecessary and unhelpful that may affect the whole system negatively or maybe lead the system to unrequired behavior [41, 57]. Fortunately, we can eliminate them inside the fogs before being sent to the mining algorithms in the cloud. This can be done by using feature selection algorithms.

1.1 Feature selection (FS) methods

Feature selection (FS) is to decide which features are more relevant and useful and which to drop without affecting the overall performance of the system [1, 54]. In electric power systems, extreme values are represented as bad records that can have an unexpected effect on loads. For example, the presence of public holidays that occurs once or twice a year in winters or summers. The demand for electrical energy will be higher than normal loads in normal times as a result of this event [58]. Because this event is not repeated, it is therefore not reliable in the learning system. Extreme values from data sets must be eliminated before applying any further processing. Whereas, these extreme values can cause noise during many data mining algorithms [22].

FS is implemented in many tasks in diverse fields including Machine Learning (ML) [34], Pattern Recognition (PR), Image Processing (IP), and multimedia. FS is a process concerning selecting relevant and informative features, so, redundant information can be avoided and ignored [1]. It majorly focuses on selecting a subset of features from the input dataset, which could effectively describe the input dataset. FS can significantly minimize the detrimental effects of noise and irrelevant characteristics on data [13, 22]. Some of the dependent features may supply no additional information. In other words, the majority of the critical information could be achieved via a few unique features that provide class discriminative information. As a result, removing the dependent features in some cases that do not correlate with the classes is essential.

There are two mainstream categories for FS and the associated taxonomy are illustrated in Table 1. They are label information and search strategy FS algorithms can be categorized concerning the search strategy into three categories (1) filter, (2) wrapper, or (3) hybrid. Filter methods depend on applying different statistical tests on each feature and ranking the features based on the score [22] and selecting the subset of features as a pre-processing step before a classification [62]. Wrapper methods select a set of features and pass them to a classifier to check the accuracy, and repeat the same process with different sets of features until the maximum accuracy is reached [10]. They also use a learning algorithm to evaluate the subsets of features according to their predictive power and accuracy [18]. A hybrid method is a mixture of filter and wrapper methods. First, features are ranked using filter methods, and then only the top scores' features are passed to the wrapper [22].

Moreover, the FS methods can be categorized into three types according to the class label information (1) supervised FS, (2) unsupervised FS, and (3) semi-supervised FS. Supervised FS methods employ labeled data for feature selection and measure the correlation of the features with the class label to determine the feature significance. To evaluate feature relevance, semi-supervised FS algorithms leverage label information from labeled data as well as the data distribution structure of both labeled and unlabeled data. Unsupervised FS methods assess feature relevance by the capability of keeping specific attributes of the data [27].

The objective of the current study is to filter the collected data selecting only the effective features of the collected data from Fogs to the cloud for the next load prediction phase. Where a perfect feature selection methodology not only improves the model prediction efficiency but also speeds up the forecasting process by considering fewer features. The main contribution of this paper is to present a new effective hybrid feature selection technique named FSBR. Figure 1 represents the framework for the proposed method which contains two phases. The first phase is the ranking phase. The features go through three different ranking methods: (1) relative weight ranking (R_{FRW}), (2) effectiveness ranking (R_{FE}), and (3) information gain ranking (R_{IG}). Then, the three ranks are passed to a fuzzy inference system to generate the final ranks based on a set of rules. The second phase is the wrapper phase, different sets of top-ranked features are passed through three different classifiers: (1) Naive Bayes, (2) Support Vector Machine (SVM), and Neural Network (NN), only one set of features is chosen based on the performances of the three classifiers.

The contributions of the current study can be summarized as follows:

- Proposing a hybrid technique in the feature selection field called FSBR.
- FSBR is to present two phases [feature ranking phase, and feature selection phase]
- feature ranking phase is to present three different ranking methods[relative weight ranking, effectiveness ranking, and information gain ranking].

Table 1 Categorization of the feature selections techniques

Mainstream Category	Category	Pros.	Cons.	Examples
Label Information	Supervised	High performance, flexibility, reliability, and accurate	Expensive, Need huge data to get high performance and a long time to implement	Self-paced regularizer (SP-regularizer) terms [16] and deep Neural network (NeuralFS) [24]
Search Strategy	Semi-Supervised	Use both labeled and unlabeled data, simple, and high efficiency	results are not stable and have low accuracy	Deep learning features [21] and local preserving logistic L-Relief [50]
	Unsupervised	Less complexity, minimized error, easy, and fast to be implemented	Less accuracy and less clear	Alternating Direction Method of Multipliers [61]
	Filter	Quick, can be scaled easily, does not depend on the classifier, has less computational complexity than wrapper methods, simply calculating the variance of feature, low time complexity, and has acceptable stability and scalability	Discard feature, dependencies, discard interaction with the classifier, and in some cases, data are unbalanced	Chi-square [7] Information gain [47, 55] Ranking-based feature selection [9]
	Wrapper	Simple, interaction with the classifier, and models feature dependencies	Risk of overfitting, classifier dependent selection, and computationally intensive	Crow search algorithm [44] and Grasshopper Optimization Algorithm [30]
	Hybrid	Merges between filter and wrapper methods, implemented by algorithms and gets high results	Classifier dependent selection	Weighted Gini index [28] and Enhanced K-Nearest Neighbor [45]

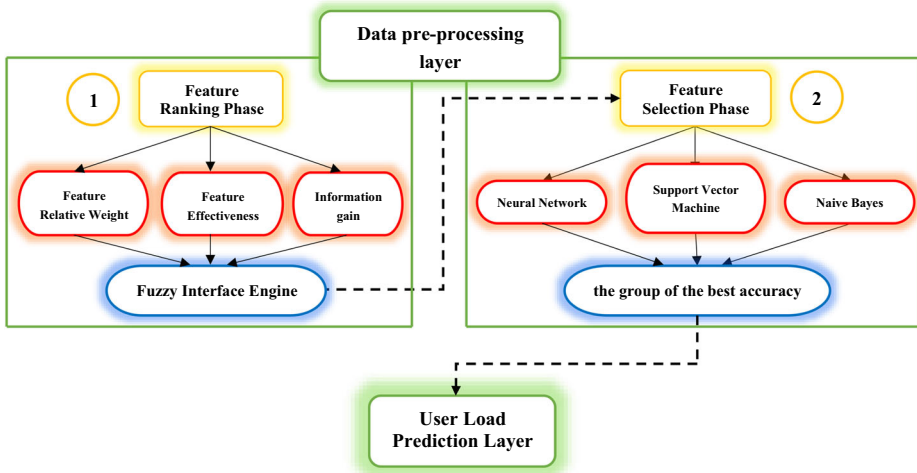


Fig. 1 The framework for the proposed method feature selection-based ranking

- feature selection phase Combining different machine learning classification algorithms(Naive Bayes, Support Vector Machine, and neural network).
- Apply the proposed method to two types of datasets.
- Comparing the current study with a set of state-of-the-art studies.

1.2 Paper organization

This paper is organized as follows: Section 2 discusses the previous efforts concerning the feature selection strategy. Section 3 presents the proposed user load forecasting strategy. Section 4 shows the experimental results. The conclusions and future work are discussed in Section 5.

2 Literature review

Initially, this section introduces a set of the previous efforts in the field of FS generally. Then, it introduces the previous efforts in the smart grid system and FS methods. Currently, there are many related works related to the concept of fog computing that discusses the difference between cloud computing [37] and edge computing technologies, its applications, emerging key technologies (i.e., communication and storage technologies), and various challenges involved in fog technology [17, 48].

Bellavista et al. [8] presented a survey on fog computing for the IoT. They illustrated the architecture of fog and some of the applications based on fog. Javadzadeh et al. [25] provided a systematic survey with a different analytical evaluation of fog computing applications in smart cities. They presented a differential approach of evolution that incorporates filter and wrapper methods into an enhanced local knowledge computational search process that is based on fuzziness principles to cope with both continuous and discrete data sets [22]. Another study by Mafarja et al. proposed an approach to solve problems in the FS using two incremental hill-climbing techniques (i.e., quick reduce and CEBARKCC). They are hybridized with the binary ant lion optimizer in a

model called HBALO [29]. Sayed et al. [44] suggested a metaheuristic optimizer, namely a chaotic crow search algorithm, to find an optimal feature subset that maximizes the classification performance and minimizes the number of selected features.

Mafarja et al. [30] presented a binary grasshopper optimization algorithm for FS problems. Binary variants were recommended and used to select the best feature subset within a wrapper-based system for classification purposes. Cilia et al. [9] presented a ranking-based approach related to the FS for handwritten character recognition. Zhu et al. [63] suggested a supervised FS algorithm to simultaneously preserve the local structure (i.e., through adaptive structure learning in a low-dimensional feature space of the original data) and the global structure (i.e., through a low-rank constraint) of the data at the same time. Bassine et al. [7] proposed an improved Arabic text classification system that used the Chi-square FS, called ImpCHI, to improve the efficiency of the classification. Verma et al. [53] applied a new hybrid approach using three FS techniques Chi-Square, Information Gain, and Principle Component Analysis, and then merged them to select the best available subset of the collected data for skin disease [60].

Ahmed et al. [3] suggested a supervised machine learning-based approach to detect a covert cyber deception assault in the state estimate with a genetic algorithm-based feature selection to improve detection accuracy. Hafeez et al. [20] provided two modules: FS like random forests and Relief-F. They were merged to create a hybrid FS algorithm to reduce the. Ahmad et al. [2] presented an artificial neural network-based day-ahead load forecasting model for smart grids, with is made up of three modules; the data preparation module, FS, and the forecast module. The data preparation module made the historical load curve compatible with the FS module to predict the future load based on the selected features. Niu et al. [35] developed a practical machine learning model based on FS using binary-valued cooperation search algorithm and parameter optimization for short-term load prediction using support vector machine. The related studies are summarized in Table 2 according to the publication year in ascending order.

Based on what was mentioned in this section, there have been researches that work only on the filtering method, while others work on the wrapper. In addition, more than one other way to feature selection was mentioned, but it had limitations such as the time, as well as computationally complex. In our proposed method, we collected the filtering method and the cover, and this gave a higher accuracy when implementing and compared to other methods, and we used easy-to-implement equations. In addition, to using equations that are easy to implement in our method.

3 The suggested feature selection strategy

An efficient forecasting model makes acceptable use of the electric loads-based data with all characteristics and also reduces its dimensionality of it. Load forecasting is classified into three types based on the time intervals. First is the short-term load forecasting that forecasts the load from a period of 24 h to one week. Short-term load forecasting makes great progress recently concerning the large data collected from smart meters [40]. The second is Medium-Term Load Forecasting, which anticipates load for a week to a year. The third is Long Term Load Forecasting, which forecasts the load from a period of one year to more than two years. With the presence of fog and the development of the centralized computing topology, we can train the load forecasting models and forecast the workloads to distributed smart meters so that consumers' raw data is handled locally and forwarded. The data to a central cloud [46].

Table 2 Related Studies Summarization

Reference	Year	Methodology	Advantages	Disadvantages	Best Accuracy	Number of Datasets
Ahmad et al. [2]	2015	Entropy-based mutual information	Decreased the average execution time and ease implementation	The model is more difficult as compared to long-term load forecasting then It may require more execution time.	97.11%	1
Ahmed et al. [3]	2018	Use a genetic algorithm	Accuracy and reduced computational complexity	The execution time is Long	93.9%	1
Zhu et al. [63]	2018	Spectral Feature Selection	Stability	a lots of complex implementation	94.66%	8
Mafarja et al. [29]	2019	QuickReduct and CEBARKCC are hybridized with the binary ant lion optimizer algorithm	Capability to balance between exploration and exploitation efficiently	This method cannot provide the optimal accuracy	97%	18
Sayed et al. [44]	2019	namely chaotic crow search algorithm	CCSA is superior compared to CSA and high performance in implementation stability	The execution time of CCSA does not reduce as possible. Additionally, It needs to be implemented on more complex science engineering problems.	100%	20
Mafarja et al. [30]	2019	Grasshopper Optimisation Algorithm	Get an accurate set of features	The method is complex in its implementation	100%	25
Bassine et al. [7]	2020	Chi-square	High performance of Arabic text classification	The method does not produce the optimal accuracy	90%	1
Verma et al. [53]	2020	A hybrid approach using Chi Square, Information Gain and Principle Component Analysis	Accuracy	Long execution time	99.67%	1
HAFEEZ et al. [20]	2020	Hybrid feature selector relief-F and random forests	Short computational time	The approach is more difficult in long-term because the implementation will take a long time	98%	1
Niu et al. [35]	2021	Binary-valued cooperation search algorithm	Short time implementation	The equations used are confused	99%	1

The suggested forecasting model [59] is built in an architectural manner that consists of three tiers. The first tier contains the IoT devices such as sensors, smart meters, monitoring systems, wireless communication devices, and demand response [6, 26]. The data is collected and sent to the fog computing layer, where it is the second tier. Fog is responsible to take data from devices such as smart meters. In the fog layer, two operations are carried out, (1) a pre-processing layer and (2) a short-term prediction model layer as shown in Fig. 2. The third tier is the cloud that contains a set of integrated data that can be used in long-term prediction. Thus, we can obtain an improved electrical grid for accurate load prediction [38].

In this paper, we focus on the fog first layer. In summary, the current study works in the data pre-processing layer using the suggested FSBR approach. In this section, we will go through the proposed feature selector in detail. It consists of two phases. First, the feature ranking phase (filter methods) using the proposed methods [relative weight ranking, effectiveness ranking, and information gain ranking]. Second, the feature selection phase (wrapper method) uses the different machine learning classification [Naive Bayes, Support Vector

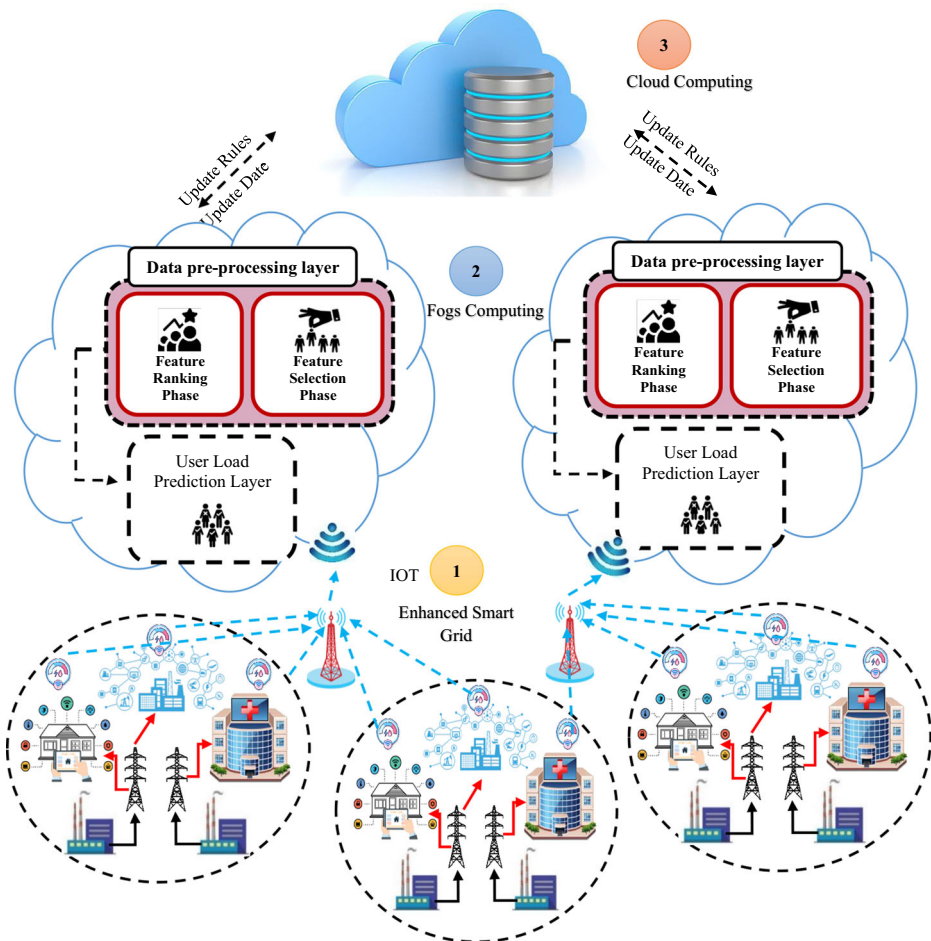


Fig. 2 User load forecasting strategy

Machine, and neural network]. As shown in Fig. 3, data is collected from smart devices and sent to fog to start the execution of the feature selection algorithm.

3.1 Feature ranking phase (FRP)

FSBR starts with the ranking phase using three different filter methods (1) Feature Relative Weight Ranking (R_{FRW}), (2) Feature Effectiveness Ranking (R_{FE}), and (3) information gain (IG) [49]. Each of the previous methods produces a different ranking for the priority and importance of the feature; so to have only one ranking for the feature, a fuzzy inference system is used. Fuzzy will take these ranks in and produce a final ranking from the three methods to give us the best features in order. so it can be used in the second phase. The feature selection

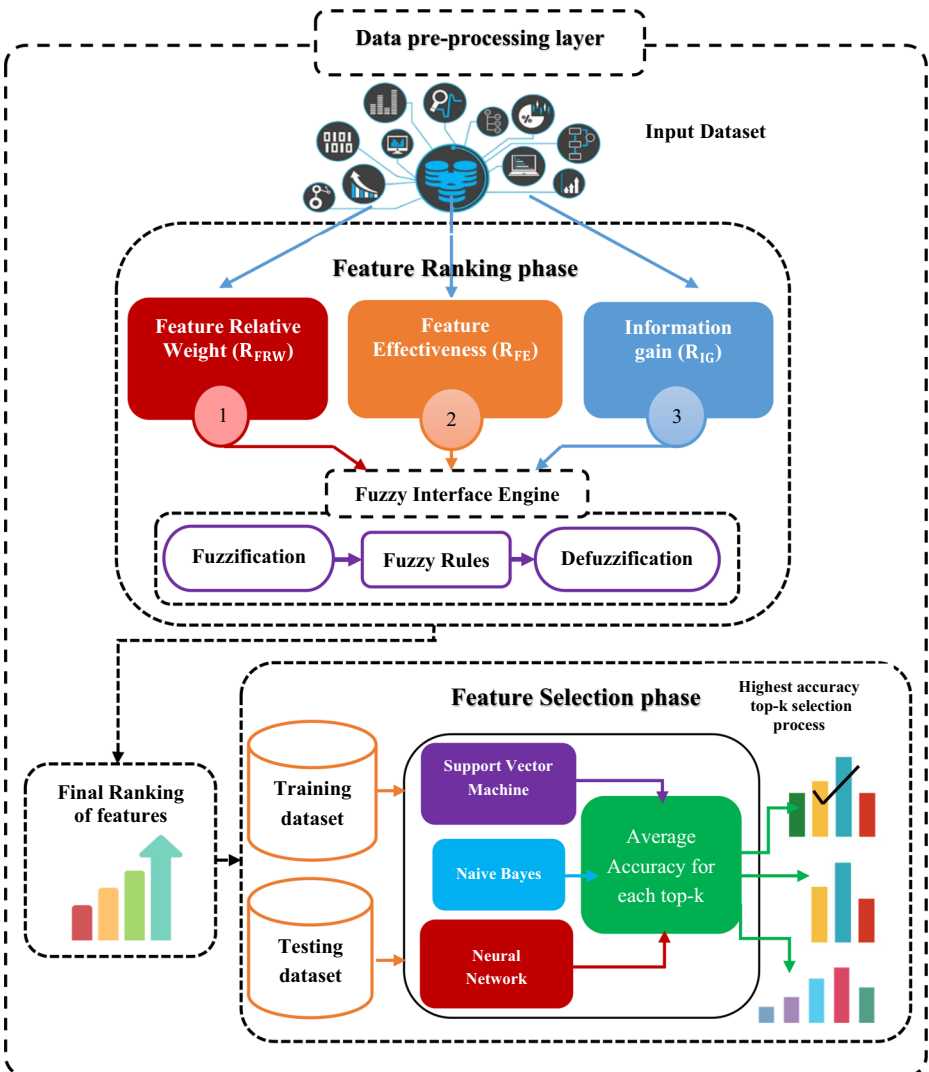


Fig. 3 The proposed feature selection phases

phase is about determining the best set of top-ranked features. This is done by determining different sets of the top-ranked features and passing them to a classifier. Based on the results of the classifier, the best set is selected. This is represented graphically in Fig. 3.

3.1.1 Feature relative weight ranking (R_{FRW})

The first-ranking method depends on the greatest feature’s impact on the output. Where the number of unique states in each feature affects the priority of the feature and the bond’s strength of output. The bond between the inputs’ features and output is a true metric to measure the importance of each feature concerning the strength of the bond. This reflects the greatest feature’s impact on the output. To achieve this, we followed an assumption that, each data column including input and output, consists of a finite number of unique states. Assume a feature F_i consists of n unique states as follows: S_1, S_2, \dots, S_n , and the output consists of m unique states as follows: O_1, O_2, \dots, O_m , where n and m are finite numbers. S_1 is repeated k times inside the feature’s column, which means the corresponding output, could be any state of output’s states and the same applies for other states. To measure the bond’s strength, we have to calculate the probability of the different output states occurring inside S_1 on its own and then for S_2, S_3 , and so on. The formulas used for this method are presented in Eqs. (1) and (2).

$$FRW(F_i, O_j) = P(S_1, O_j) + P(S_2, O_j) + \dots + P(S_n, O_j) \tag{1}$$

$$R_{FRW}(F_i) = |\text{diff}(\overline{FRW(F_i, O_1)}, \overline{FRW(F_i, O_2)}, \dots, \overline{FRW(F_i, O_m)})| \times n \tag{2}$$

Features Ranking Using R_{FRW} Algorithm

Inputs:

TDS = (D, F); the training dataset
 O = output set = { O_H, O_L }
 m = |LTDI| or |D|; number of items or labels in the training dataset
 i = |F|; number of features in training and testing dataset

Outputs:

RF1 = $FR1_1, \dots, FR1_n$; list of ranking features in the dataset

Steps:

1. Determine the unique state for all Training items at every feature in TDS
2. For every F_i in F
3. Calculate unique S_n , where S_n is the set the unique state
4. Calculate occurrence of each state s_j in S_n
5. End for
6. For every F_i in F
7. For every O_j in O
8. Calculate FRW of F_i with every O_j :-

$$P(F_i, O_j) = \frac{S_n}{\text{total number } S_n \text{ in } O_j}$$
9. End for
10. Calculate FRW of F_i with all classes:-

$$FRW(F_i, O_j) = \sum_1^S \frac{S_n}{\text{total number } S_n \text{ in } O_j}$$
11. End for
12. // calculate the rank feature relative weight
13. For every F_i in F

$$R_{FRW}(F_i) = |\text{Diff}(FRW(F_i, O_1), FRW(F_i, O_2), \dots, FRW(F_i, O_m))| * n$$
14. End for
15. RF1 = contains list of ranking features in R_{FRW}

Algorithm Parameters

TDS	Training data set contents of training items and its features, TDS = (D, F).
O	Set of the class
F	Features of training or testing items, $F=f_1; \dots; f_x$.
D	Training items, $D=I_1; \dots; I_m$.
m	$m = LTDI $ or $ D $; number of items or Labels in Training data set
i	$i = F $; number of features in Training and testing data set
F_i	Feature belongs to unique features from D
O_j	Class belongs to unique classes from LTDI
S_n	Number of states in the features
$FRW(F_i, O_j)$	the probability of states in the features on the class
R_{FRW}	feature relative weight for every F_j according to labels LTDI.
n	number of unique states of feature F_i

Fig. 4 Algorithm1: Features Ranking using R_{FRW} Algorithm

Table 3 A Sample Dataset with 15 Records

#	Features						Output
	Time	Season	Weather	Weekday	Holiday	Event	Load
1	T1	Summer	Hot	Yes	No	No	High
2	T1	Summer	Fine	Yes	No	Yes	High
3	T1	Fall	Fine	Yes	No	No	Low
4	T1	Fall	Fine	Yes	Yes	No	High
5	T1	Winter	Cool	Yes	No	No	Low
6	T1	Winter	Fine	Yes	No	Yes	High
7	T2	Summer	Hot	Yes	No	No	High
8	T2	Spring	Fine	Yes	No	No	Low
9	T2	Fall	Fine	Yes	Yes	Yes	High
10	T2	Winter	Cool	Yes	No	No	Low
11	T2	Winter	Cool	Yes	No	Yes	High
12	T3	Summer	Hot	No	No	No	High
13	T3	Winter	Cool	No	No	No	Low
14	T3	Winter	Cool	No	Yes	Yes	High
15	T3	Spring	Fine	No	No	Yes	High

where i is the feature’s index, j is the output’s state’s index, $P(S_n, O_j)$ is the probability of the output O_j to occur when the input’s state is S_n . where $R_{FRW}(F_i)$ is the feature relative weight ranking for feature F_i .

The R_{FRW} is calculation is presented in Algorithm 1 (Fig. 4). To simplify the illustration of this method, a sample dataset (Table 3) of 15 items (6 features and 1 output) is used for the explanation where T1 indicates (Saturday, Sunday, Monday), T2 indicates; (Tuesday, Wednesday, Thursday), and T3 indicates (Friday) are respectively. The employed states of the features are presented in Table 4. illustrates how to execute the R_{FRW} method on Table 5.

Table 4 The Employed States of the Features of the Sample Dataset

Column	Number of Unique States (n)	Unique States	Occurrence
Time	3	T1	6
		T2	5
		T3	4
Season	4	Summer	4
		Fall	3
		Winter	6
		Spring	2
Weather	3	Hot	3
		Fine	7
		Cool	5
Weekday	2	Yes	11
		No	4
Holiday	2	Yes	3
		No	12
Event	2	Yes	6
		No	9
Output	2	High (O_1)	10
		Low (O_2)	5

Table 5 Illustration of the Relative Weight Ranking Method on an Example

Step	Description	Sample results	Full results																					
1	Calculate FRW(F_i, O_j) using Eq. (1) for each class	$FRW(F_1, O_1) = \frac{4}{6} + \frac{3}{5} + \frac{3}{4} = 2.016$ $FRW(F_1, O_2) = \frac{2}{6} + \frac{2}{5} + \frac{1}{4} = 0.9833$ The same is applied for other features, the result of F _i are reported in Table A	<table border="0"> <tr> <td>F_i</td> <td>FRW(F_i, O_H)</td> <td>FRW(F_i, O_L)</td> </tr> <tr> <td>F₁</td> <td>2.016</td> <td>0.9833</td> </tr> <tr> <td>F₂</td> <td>2.100</td> <td>0.9000</td> </tr> <tr> <td>F₃</td> <td>2.125</td> <td>0.8800</td> </tr> <tr> <td>F₄</td> <td>1.386</td> <td>0.6140</td> </tr> <tr> <td>F₅</td> <td>1.583</td> <td>0.4170</td> </tr> <tr> <td>F₆</td> <td>1.444</td> <td>0.5555</td> </tr> </table>	F_i	FRW(F_i, O_H)	FRW(F_i, O_L)	F₁	2.016	0.9833	F₂	2.100	0.9000	F₃	2.125	0.8800	F₄	1.386	0.6140	F₅	1.583	0.4170	F₆	1.444	0.5555
F_i	FRW(F_i, O_H)	FRW(F_i, O_L)																						
F₁	2.016	0.9833																						
F₂	2.100	0.9000																						
F₃	2.125	0.8800																						
F₄	1.386	0.6140																						
F₅	1.583	0.4170																						
F₆	1.444	0.5555																						
2	R_{FRW} calculation using Eq. (2)	$F_1 = 2.0167 - 0.9833 * 3 = 3.1002$ The same is applied for other features, the results are reported in Table B	<table border="0"> <tr> <td>F_i</td> <td>R_{FRW}(F_i, O_j)</td> </tr> <tr> <td>F₁</td> <td>3.1</td> </tr> <tr> <td>F₂</td> <td>3.6</td> </tr> <tr> <td>F₃</td> <td>3.75</td> </tr> <tr> <td>F₄</td> <td>1.5452</td> </tr> <tr> <td>F₅</td> <td>2.333</td> </tr> <tr> <td>F₆</td> <td>1.78</td> </tr> </table>	F_i	R_{FRW}(F_i, O_j)	F₁	3.1	F₂	3.6	F₃	3.75	F₄	1.5452	F₅	2.333	F₆	1.78							
F_i	R_{FRW}(F_i, O_j)																							
F₁	3.1																							
F₂	3.6																							
F₃	3.75																							
F₄	1.5452																							
F₅	2.333																							
F₆	1.78																							

3.1.2 Feature effectiveness ranking

The second-ranking method is driven by the popular Naïve Bayes classifier that has presented in Eq. (3).

$$P(O | F) = \frac{P(F | O) \times P(O)}{P(F)} \tag{3}$$

The newly proposed ranking technique inherits the same formula but instead of applying it once, using only one state of each feature to get the probability of output, we used it on each feature’s state to get The effect of the feature over other features. Where $P(O | F)$ is the probability of target O to happen if given attribute F , $P(F | O)$ is the conditional probability of F given O , $P(O)$ is the probability of the class, and $P(F)$ is the probability of the attribute. It is used as a classifier by giving attributes as input and getting a probability as an output. The higher probability we get, the higher chance the output to occur. We have driven a new ranking technique using this probability based on the different states of each feature as mentioned in the first ranking method.

The newly proposed ranking technique inherits the same previous formula but instead of applying it once, using only one state of each feature to get the probability of output, we used it on each feature’s state to get the probability of each output’s state as presented in Eq. (4).

$$P(O_j | F_i) = \sum_{i=1}^n P(O_j | S_i) \tag{4}$$

where $P(O_j | F_i)$ is the probability to get output state O_j if given feature F_i , $P(O_j | S_i)$ is the probability to get output state O_j if given feature’s state S_i , n is the number of feature’s states, and O_j is the required output’s state. Then the following formula is used to get a final rank that represents the whole feature’s column as presented in Eq. (5).

$$R_{FE}(F_i) = \text{diff}\left(P(O_1 | F_i), P(O_2 | F_i), \dots, P(O_m | F_i)\right) \tag{5}$$

where $R_{FE}(F_i)$ is the feature effectiveness of the feature’s column F_i , and $P(O_1 | F_i)$ is the probability to get output state O_1 if given feature F_i . Rank Feature Effectiveness is illustrated in

Features Ranking Using R_{FE} Algorithm

Inputs:

TDS = (D, F); Training dataset
 O = output set = { O_H, O_L }
 LTDI = $l_{t_{i1}}, \dots, l_{t_{im}}$; labels of training items
 m = |LTDI| or |D|; number of items or labels in training dataset
 i = |F|; number of features in training and testing dataset

Outputs:

RF2 = $FR1_1, \dots, FR1_n$; list of ranking features in the dataset

Steps:

1. Determine the probability of class
 // Calculate effectiveness of feature F_i on the class LTDI
2. For every F_i in F
3. $E(F_i, O_j) = P(F_i | O_j) * P(O_j)$
4. End for
5. For every F_i in F
6. $P(F_i) = \frac{\text{total item } S_n \text{ in } F_i}{\text{total item in } O}$
7. End for
 // Calculate the rank feature effectiveness
8. For every F_i in F
9. $P(O_j | F_i) = \sum_{i=1}^n P(O_j | S_i)$
10. End for
11. For every F_i in F
12. For every O_j in O
13. $R_{FE}(F_i) = \text{Diff}(P(O_1 | F_i), P(O_2 | F_i) \dots P(O_m | F_i))$
14. End for
15. End for
16. RF2 = contains list of ranking features in R_{FE}

Algorithm Parameters	
TDS	Training data set contents of training items and its features, TDS = (D, F).
LTDI	Labels of training items, LTDI = $l_{t_{i1}}, \dots, l_{t_{im}}$;
F	Features of training or testing items, F= f_1, \dots, f_n .
D	Training items, D= I_1, \dots, I_m .
m	m = LTDI or D ; number of items or Labels in Training data set
i	i = F ; number of features in Training and testing data set
F_i	Feature belongs to unique features from D
O_j	Class belongs to unique classes from LTDI
P(F_i)	Marginal probability density function for F_x feature.
P(O_j)	Marginal probability density function for C_k class.
E(F_i, O_j)	effectiveness of feature F_x on the class LTDI
P($F_i O_j$)	probability of predictor(F_x) given class(C_k).
R_{FE}	Feature Effectiveness for every F_j according to labels LTDI.

Fig. 5 Algorithm 2: Features Ranking Using R_{FE} Algorithm

Algorithm 2 (Fig. 5). To simplify the previous formulas and the ranking technique, the dataset shown in Table 3 is used a gain to simplify the illustration as show in Table 6.

3.1.3 Information gain

Information gain [15] is used as a third-ranking method with the previous two methods where it measures the mutual dependence between two variables such as the dependence between an input’s feature and the output. From our context, the higher information gain, the strongest relation between input’s feature and the output. Hence, we aim to find which features have the highest values with the output. Equation 6 shows the used formula to calculate IG.

$$IG(y, X) = E(y) - E(X | y) \tag{6}$$

where $E(y)$ is the target’s entropy and $E(X | y)$ is the measure of the entropy of target y given variable X . The term Entropy represents the measure of uncertainty or disorder, which has the formula shown in Eq. 7.

$$E(y) = - \sum_{i=1}^m P(y_i) \times \text{Log}_2(P(y_i)) \tag{7}$$

where m is the number of target’s classes, $P(y_i)$ is the probability of class y_i , and $\text{Log}_2(P(y_i))$ is the logarithmic value of the class’s probability. $E(X | y)$ can be calculated from:

Table 6 The Followed Steps of Rank Feature Effectiveness

Step	Description	Results				
1	Calculate the intersection features to class F_i $P(F_{i,S_n} O_j) = \frac{\text{total item } S_n \text{ in } F_i}{\text{total item in } O_j}$	F_1	High	Low	$P(F_{1,S_n} O_H)$	$P(F_{1,S_n} O_L)$
		T_1	4	4	$\frac{4}{10} = 0.4$	$\frac{2}{5} = 0.4$
		T_2	3	3	$\frac{3}{10} = 0.3$	$\frac{2}{5} = 0.4$
		T_3	3	3	$\frac{3}{10} = 0.3$	$\frac{1}{5} = 0.2$
		Total	10	10		
2	Calculate $E(F_i, O_j)$ $E(F_i, O_j) = P(F_{i,S_n} O_j) * P(O_j)$	$P(O_j)$	0.666	0.333	$P(O_H)$	$P(O_L)$
		$E(F_i, O_j)$	$0.4*0.666=0.2664$	$0.4*0.333=0.1332$	$0.3*0.666=0.1998$	$0.4*0.333=0.1332$
			$0.3*0.666=0.1998$	$0.2*0.333=0.0666$		
			The same calculations are applied to all features			
3	Calculate $P(F_i)$ $P(F_i) = \frac{\text{total item } S_n \text{ in } F_i}{\text{total item}}$	$P(F_1, S_1)$	$= \frac{6}{15} = 0.4$			
		$P(F_1, S_2)$	$= \frac{5}{15} = 0.33$			
		$P(F_1, S_3)$	$= \frac{4}{15} = 0.266$			
4	Calculate $R_{FE}(F_i, O_j)$ $P(O_j F_i) = \sum_{i=1}^n P(O_j S_i)$	S_1	$\frac{0.2664}{0.4} = 0.66$	$P(O_H S_i)$	$P(O_L S_i)$	$\frac{0.1332}{0.4} = 0.33$
		S_2	$\frac{0.1998}{0.333} = 0.6$			$\frac{0.1332}{0.333} = 0.4$
		S_3	$\frac{0.1998}{0.266} = 0.75$			$\frac{0.0666}{0.266} = 0.25$
		$P(O_j F_i)$	2.016			0.9833
5	Feature Effectiveness for other features in the dataset.	F_i	$R_{FE}(F_i, O_H)$			$R_{FE}(F_i, O_L)$
		F_1	2.016			0.9833
		F_2	2.1			0.9
		F_3	2.114			0.88
		F_4	1.386			0.613
		F_5	1.583			0.416
		F_6	1.444			0.555
6	The final results $R_{FE}(F_i) = \text{diff}(P(O_1 F_i), P(O_2 F_i), \dots, P(O_m F_i))$	F_i	$R_{FE}(F_i, O_j)$			
		F_1	1.03			
		F_2	1.2			
		F_3	1.23			
		F_4	0.77			
		F_5	1.17			
		F_6	0.89			

$$E(X | y) = \sum_{i=1}^m P(y_i) \times \sum_{j=1}^n P(X_j | y_i) \times \text{Log}_2(P(X_j | y_i)) \tag{8}$$

To measure a feature’s IG first, we measure the whole dataset’s output entropy. Then, we measure the entropy of the output but for a given input’s feature. Finally, we subtract both values to get the final value of the IG. Table 7 shows the values of IG for each feature from the dataset presented in Table 3. Thus, we have obtained the third and final rank of the proposed types of ranks.

Table 7 Third Rank Values using the IG Method

Feature	IG
F1	0.011
F2	0.195
F3	0.192
F4	0.008
F5	0.134
F6	0.324

3.2 Fuzzy Interface engine

The previous ranking methods will lead to three different ranking tables. Hence, somehow it is necessary to merge them into only one. A fuzzy inference system is the best among alternatives to do that. Fuzzy is based only on simple IF-THEN rules merged with fuzzy logic operators (e.g., AND and OR) to enhance the decision-making which is much similar to humans’ reasoning. Fuzzy inference is a process to map the input into an output using fuzzy logic [43]. The previous process goes as follows (1) the input of crisp values is converted into fuzzy quantities, (2) then it goes through the fuzzy rules and fuzzy memberships to generate an output, and (3) the output is in a form of a fuzzy set that needs to be defuzzified to get the output of crisp values once again.

The input to the fuzzy system is the crisp values as shown in Table 8. They need to be converted into fuzzy sets using the memberships shown in Fig. 6. To generate the corresponding memberships for the three ranking methods, we need to determine the α , β , and γ values as presented in Eqs. 9 to 11 [43].

$$\alpha = \frac{0.5 \times \sum_{x=1}^n (value)_x}{n} \tag{9}$$

$$\beta = \alpha \times 2 \tag{10}$$

Table 8 Crisp Input Values of the Fuzzy Inference Engine

Crisp Input Fuzzy			
Feature	R _{FRW}	R _{FE}	R _{IG}
F ₁	3.1	1.03	0.011
F ₂	3.6	1.2	0.195
F ₃	3.75	1.23	0.192
F ₄	1.5452	0.77	0.008
F ₅	2.333	1.17	0.134
F ₆	1.78	0.89	0.324
$\sum_{x=1}^n (value)_x$	16.108	6.29	0.87

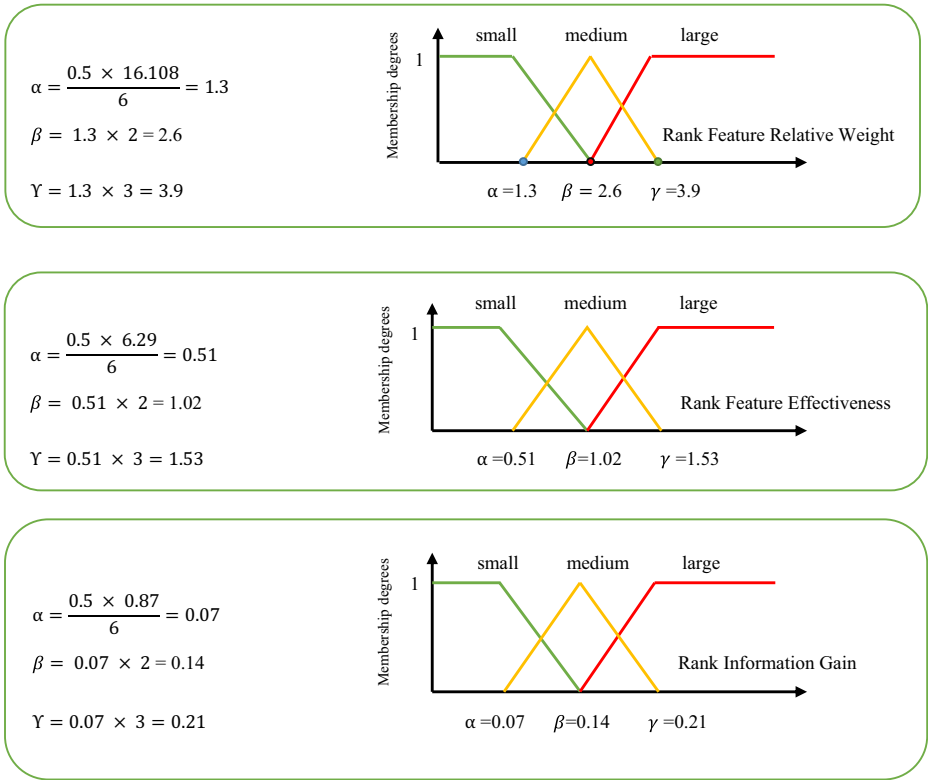


Fig. 6 The Membership Function

$$\Upsilon = \alpha \times 3 \tag{11}$$

After calculating the three values of α , β , and Υ , each value from each ranking is converted into fuzzy input that can be small (S), medium (M), or large (L) by using the Eqs. 12 to 14.

$$\mu(x)_{small} = \begin{cases} 1 & x \leq \alpha \\ \frac{\beta-x}{\beta-\alpha} & \alpha < x \leq \beta \\ 0 & x > \beta \end{cases} \tag{12}$$

$$\mu(x)_{medium} = \begin{cases} 0 & x \leq \alpha \\ \frac{x-\alpha}{\beta-\alpha} & \alpha < x \leq \beta \\ \frac{\Upsilon-x}{\Upsilon-\beta} & \beta < x \leq \gamma \\ 0 & x > \gamma \end{cases} \tag{13}$$

$$\mu(x)_{large} = \begin{cases} 0 & x \leq \beta \\ \frac{x-\beta}{\Upsilon-\beta} & \beta < x \leq \gamma \\ 1 & x > \gamma \end{cases} \tag{14}$$

where n is the number of the input values for the methods, value is the ranking of the feature x related to a particular method.

Consequently, the fuzzy rule-based input is the fuzzification process’ output. A set of rules are considered here in the form; if (X is A) AND (Y is B) THEN (Z is C), where A, B, and C represent input variables (R_{FRW} , R_{FE} , and R_{IG}) and A, B, and C represent the corresponding linguistic variables (e.g., small, medium, and large). The first part of the rule (before THEN) is called “antecedent”. The second part (after THEN) is called “consequent”. These input fuzzy sets go through if-then rules to determine the output. In this paper, there are 27 different rules used to determine the output as shown in Table 9. Then the output goes into a defuzzify process to get crisp values back representing the final ranking.

Defuzzification can be applied using different methods such as max-min, max criterion, center-of-gravity (COG), and the mean of maxima [5, 39, 43]. The max-min depends on choosing a min operator for the conjunction in the premise of the rule as well as for the implication function and the max aggregation operator [5]. Consider a simple case of two elements of evidence per rule, the corresponding rules will be:

$$\mu_M = \overbrace{\max}^{aggregation} \left(\underbrace{\min}_{implication} (\mu_{k_{j1}}, \mu_{k_{j2}}) \forall j \in \{1, 2, 3, \dots, N\} \right) \tag{15}$$

This yields to:

$$\mu_M = \max(\min(\mu_{k_{11}}, \mu_{k_{12}}), \min(\mu_{k_{21}}, \mu_{k_{22}}), \dots, \min(\mu_{k_{N1}}, \mu_{k_{N2}})) \tag{16}$$

Table 9 The Used Fuzzy Rules

ID	R_{FRW}	R_{FE}	R_{IG}	Rule (Output)	ID	R_{FRW}	R_{FE}	R_{IG}	Rule (Output)
1	S	S	S	S	15	M	M	L	M
2	S	S	M	S	16	M	L	S	M
3	S	S	L	S	17	M	L	M	M
4	S	M	S	S	18	M	L	L	L
5	S	M	M	M	19	L	S	S	S
6	S	M	L	M	20	L	S	M	M
7	S	L	S	S	21	L	S	L	L
8	S	L	M	M	22	L	M	S	M
9	M	L	L	M	23	L	M	M	M
10	M	S	S	S	24	L	M	L	L
11	M	S	M	M	25	L	L	S	M
12	M	S	L	M	26	L	L	M	L
13	M	M	S	M	27	L	L	L	L
14	M	M	M	M					

The COG is the most popular one [5] and this is the used method in the current study. This method is identical to the formula for calculating the center of gravity in physics. The membership function, In our case, is bounded by the weighted average of the membership function or the COG of the area. Defuzzification can be accomplished by the output membership function shown in step 3 in Table 10. Assuming $\alpha = 3, \beta = 6,$ and $\gamma = 9$ related to Eqs. 9 to 11. In Algorithm 3 (Fig. 7), the final ranking using the fuzzy interface engine of the fuzzy rank is used to get results of the implemented fuzzy rank (R) method. An example is illustrated in Table 10.

Table 10 The Followed Steps of the Fuzzy Rank Method

Step	Description	Result																														
1	Find the fuzzy set membership values for each input $(R_{FRW}=3.1, R_{FE}=1.03, R_{IG}=0.011)$	<p>For R_{FRW}: Related to Figure 6 and Equations (7-9) $\mu(x)_{small} (R_{FRW} = 3.1) = 0$ $\mu(x)_{medium} (R_{FRW} = 3.1) = 0.68$ $\mu(x)_{large} (R_{FRW} = 3.1) = 0.318$</p> <p>For R_{FE}: Related to Figure 6 and Equations (7-9) $\mu(x)_{small} (R_{FE} = 1.03) = 0$ $\mu(x)_{medium} (R_{FE} = 1.03) = 0.984$ $\mu(x)_{large} (R_{FE} = 1.03) = 0.0156$</p> <p>For R_{IG}: Related to Figure 6 and Equations (7-9) $\mu(x)_{medium} (R_{IG}=0.011) = 0$ $\mu(x)_{small} (R_{IG}=0.011) = 1$ $\mu(x)_{large} (R_{IG}=0.011) = 0$</p>																														
2	Calculate the fuzzy outputs for each rule.	<table border="1"> <thead> <tr> <th>ID</th> <th>R_{FE}</th> <th>R_{FE}</th> <th>R_{IG}</th> <th>Rule Output</th> <th>Output Membership</th> </tr> </thead> <tbody> <tr> <td>13</td> <td>M</td> <td>M</td> <td>S</td> <td>M</td> <td>Min (0.68,0.984,1) =0.68</td> </tr> <tr> <td>16</td> <td>L</td> <td>L</td> <td>S</td> <td>M</td> <td>Min (0.68,0.0156,1) =0.0156</td> </tr> <tr> <td>22</td> <td>M</td> <td>M</td> <td>S</td> <td>M</td> <td>Min (0.318,0.984,1) =0.318</td> </tr> <tr> <td>25</td> <td>L</td> <td>L</td> <td>S</td> <td>M</td> <td>Min(0.318,0.0156,1) =0.0156</td> </tr> </tbody> </table>	ID	R_{FE}	R_{FE}	R_{IG}	Rule Output	Output Membership	13	M	M	S	M	Min (0.68,0.984,1) =0.68	16	L	L	S	M	Min (0.68,0.0156,1) =0.0156	22	M	M	S	M	Min (0.318,0.984,1) =0.318	25	L	L	S	M	Min(0.318,0.0156,1) =0.0156
		ID	R_{FE}	R_{FE}	R_{IG}	Rule Output	Output Membership																									
		13	M	M	S	M	Min (0.68,0.984,1) =0.68																									
		16	L	L	S	M	Min (0.68,0.0156,1) =0.0156																									
		22	M	M	S	M	Min (0.318,0.984,1) =0.318																									
25	L	L	S	M	Min(0.318,0.0156,1) =0.0156																											
So that the output will be: Medium with membership =																																
MAX (0.68, 0.0156, 0.318, 0.015) = 0.68																																
3	Defuzzification Use the center of gravity method: $COG = \frac{\sum \mu(x) * x}{\sum \mu(x)}$	<p>$COG = \frac{(4+5+6+7+8)0.68}{(0.68*5)} = 6$</p> <p>Output: Ranking of F_1 equals 6</p>																														
4	Repeating the previous three steps for F_2, F_3, F_4, F_5, F_6 (the other input in table8).	<table border="1"> <thead> <tr> <th>F_i</th> <th>$R(F_i, O_j)$</th> </tr> </thead> <tbody> <tr> <td>F_1</td> <td>6</td> </tr> <tr> <td>F_2</td> <td>7.41</td> </tr> <tr> <td>F_3</td> <td>7.13</td> </tr> <tr> <td>F_4</td> <td>3.44</td> </tr> <tr> <td>F_5</td> <td>5.51</td> </tr> <tr> <td>F_6</td> <td>4.81</td> </tr> </tbody> </table>	F_i	$R(F_i, O_j)$	F_1	6	F_2	7.41	F_3	7.13	F_4	3.44	F_5	5.51	F_6	4.81																
		F_i	$R(F_i, O_j)$																													
		F_1	6																													
		F_2	7.41																													
		F_3	7.13																													
		F_4	3.44																													
		F_5	5.51																													
F_6	4.81																															

Final Features Ranking Using Fuzzy rankAlgorithm

Inputs:

Input ranks = {R_{FRW}, R_{FE}, R_{IG}}

O = output set = {O_H, O_L}

i = |F|; number of features in training and testing dataset

Outputs:

R(Fi) = RF₁, ..., RF_i; list of fuzzy ranking features in the dataset

Steps:

1. Determine the sum of value in each rank method
2. Value = $\sum_1^i F_i$
3. For every F_i in F
4. Calculate α β γ values
5. $\alpha = \frac{0.5 \times \sum_{x=1}^n (value)_x}{n}$
6. $\beta = \alpha \times 2$
7. $\gamma = \alpha \times 3$
8. End for
9. For every F_i in F
10. Calculate *small, medium, large* for each ranking
11. If $x \leq \alpha$
12. $M(x)_{small} = 1$
13. If $\alpha < x \leq \beta$
14. $M(x)_{small} = \frac{\beta-x}{\beta-\alpha}$
15. If $x > \beta$
16. $M(x)_{small} = 0$
17. If $x \leq \alpha$
18. $M(x)_{medium} = 0$
19. If $\alpha < x \leq \beta$
20. $M(x)_{medium} = \frac{x-\alpha}{\beta-\alpha}$
21. If $\beta < x \leq \gamma$
22. $M(x)_{medium} = \frac{\gamma-x}{\gamma-\beta}$
23. If $x > \gamma$
24. $M(x)_{medium} = 0$
25. If $x \leq \beta$
26. $M(x)_{large} = 0$
27. If $\beta < x \leq \gamma$
28. $M(x)_{large} = \frac{x-\beta}{\gamma-\beta}$
29. If $x > \gamma$
30. $M(x)_{large} = 1$
31. End for
32. For every F_i in F
33. Calculate the fuzzy rule output
34. Calculate μ_M
35. $\mu_M = \max(\min(\mu_{k_{11}}, \mu_{k_{12}}), \min(\mu_{k_{21}}, \mu_{k_{22}}), \dots, \min(\mu_{k_{N1}}, \mu_{k_{N2}}))$
36. $COG = \frac{\sum \mu(x) \cdot x}{\sum \mu(x)}$
37. End for
38. R(Fi) = contains list of ranking features in R_{FRW}

Algorithm Parameters	
Input ranks	Crisp input features values for each rank
O	Set of the class
F	Features of training or testing items, F=f ₁ ;.....;f _x .
i	i = F ; number of features in Training and testing data set
F_i	Feature belongs to unique features from dataset
R_{FRW}	The first rank method
R_{FE}	The second rank method
R_{IG}	The third rank method
α, β, γ	Range value in membership
M(x)_{small}, M(x)_{medium}, M(x)_{large}	Set membership value
COG	Center of gravity

Fig. 7 Algorithm 3: final ranking using fuzzy interface engine

3.3 Feature selection phase (FSP)

The results of the previous phase are the ranks of all features ranked from the most effective features to the least ones. This is suitable as we can drop the least effective ones to reduce the computations and complexity, ending with the most effective features but they could be many, for big systems they could be hundreds or thousands. Therefore, the question is; **What is the least** amount of the most effective features that could be enough to operate as if we had used every feature from the features set? this question will be answered in this section. A wrapper

phase is a tray-and-error phase where we use different classifiers with a different number of top-ranked features to end with the least number of features that could do the same job as the whole features set.

In this work, we have used three different classifiers (1) Simple Neural Network, (2) Naïve Bayes, and (3) Support Vector Machine. First, we have to determine the accuracy of each classifier to compare the results with the different combinations of top-ranked features. Then, we determine the average accuracy in each combination, and only one set of these features is chosen based on the highest average accuracy. As an explanation, the dataset described in Table 3 is divided into 15 items for training and 10 items for testing.

First, it is time to test the top-ranked sets. The top-ranked features for this sample are [Season, Weather, Time, Weekday, Holidays, and Events]. Five different sets were made, Top 1 to Top 6, where Top 1 contains only the first top-ranked feature, which is the Season, and Top 5 contains top-five-ranked features. Results are shown in Table 11. Nevertheless, for the sake of simplicity, fewer computations, and faster decision-making; Top 3 is better than Top 5, as only three features could do the same job as the whole features set.

During our work, as a summary, we discussed the first phase that is called the feature ranking phase (FRP) by applying the filter feature selection methods. The final features ranking are the outputs of the fuzzy method. Illustrative calculations are performed on a sample dataset (Table 3). The second stage is called the feature selection phase (FSP) will apply a wrapper methodology feature selection method to select the best features from x feature ranking according to three classifiers (1) neural network, (2) naive Bayes, and (3) support vector machine. Based on the output fuzzy, the order of features is [Season, Weather, Time, Holidays, Events, Weekday]. First, we will test each classifier on the data multiple times. The first experiment calculates the accuracy from the first feature TOP1 [Season] then calculates the accuracy by adding the following feature TOP2 [Season, Weather]. Thus, We continue to add the features gradually, then find six results in each classifier. With the imposition of the values as shown in Table 11, we find that at the TOP3 and TOP5 features. We have the average accuracy of TOP3 is 89.66% and of TOP5 is 88.33%. When calculating the average accuracy, we found that TOP3 features are the best for sake of simplicity.

4 Evaluation and results

The current section shows the experiments, reported results, and the corresponding discussions. The experiments are performed on Windows 10 using the Python programming language. The used packages are NumPy, Pandas, Keras, and Scikit-Learn. The environment had an Intel Core i7 processor with a RAM of 6 GB.

Table 11 The Accuracy Results

	NN (%)	NB (%)	SVM (%)	Average (%)
TOP1	88	84	85	85.7
TOP2	90	85	86	87
TOP3	95	87	87	89.7
TOP4	90	84	86	86.7
TOP5	94	86	85	88.33
All features (TOP6)	93	83	84	86.7

4.1 Datasets

The current study uses two datasets. The first is the EUNITE dataset while the second is the USPS dataset.

4.1.1 EUNITE dataset

European Network on Intelligent Technologies or EUNITE [15] is a dataset that contains electrical loads for Eastern Slovakian Electricity Corporation during the period between January 1, 1997, and December 31, 1998. It was in the form of four columns that contain daily information as follows (1) date, (2) temperature, (3) holiday, and (4) load. The holiday column represents if the corresponding day meets an annual holiday or not (e.g., Christmas and Easter). The first three columns are the features while the last one (load) is the desired output, but for a complete test and representation of our proposed method, we have added three more columns based on the day's date. These columns are (1) weekday, (2) event, and (3) season. The event represents if there any occasional events happened. The resultant modified dataset consists of 730 samples and seven columns (i.e., six columns as features and one column as the desired output). Samples from the modified dataset are shown in Table 12.

The data in its current form cannot be used, neither for the filter phase nor for the selection phase, because there are different forms for each column and a wide range for numerical columns temperature and load. Hence, it is necessary to go through a preprocessing stage to reform the data in an appropriate numerical form. This stage converts the date column into a number that represents a day in the week. It also converts the holiday, weekday, and events columns into 1 or 0 where 1 is Yes or 0 is No. Finally, it normalizes the temperature and load columns. The temperature range is set to $[-1, 1]$ and the load range is set to $[0, 1]$. Table 13 shows the results after the preprocessing stage.

Table 12 Samples from the Modified EUNITE Dataset

Date	Temperature (°F)	Holiday	Weekday	Event	Season	Load (kW)
1/1/1997	-7.6	Yes	Yes	No	Winter	32,715
1/2/1997	-6.3	No	Yes	No	Winter	34,293
1/3/1997	-3.2	No	No	No	Winter	33,221
1/4/1997	-5.4	No	No	No	Winter	31,815
4/7/1997	1.4	No	Yes	Yes	Spring	31,222
4/8/1997	1.7	No	Yes	Yes	Spring	32,196
4/9/1997	4.7	No	Yes	No	Spring	31,704
4/10/1997	8.3	No	Yes	No	Spring	30,391
7/4/1997	22.7	No	Yes	No	Summer	24,213
7/5/1997	19.7	Yes	No	No	Summer	22,124
7/6/1997	16.6	No	No	No	Summer	21,262
7/7/1997	20.7	No	Yes	No	Summer	23,312
11/1/1998	4.6	No	No	No	Fall	30,236
11/2/1998	5.7	Yes	No	Yes	Fall	28,612
11/3/1998	3.7	Yes	Yes	No	Fall	30,964

Table 13 Samples from the Preprocessed Modified EUNITE Dataset

Day Index	Temperature (°F)	Holiday	Weekday	Event	Season	Load (kW)
2	-0.6	1	1	0	3	0.7
3	-0.6	0	1	0	3	0.7
5	-0.5	0	0	0	3	0.7
6	-0.5	0	0	0	3	0.6
0	-0.3	0	1	1	1	0.6
1	-0.3	0	1	1	1	0.6
2	-0.2	0	1	0	1	0.6
3	0	0	1	0	1	0.6
4	0.5	0	1	0	2	0.3
5	0.4	1	0	0	2	0.3
6	0.3	0	0	0	2	0.2
0	0.4	0	1	0	2	0.3
5	-0.2	0	0	0	0	0.6
6	-0.1	0	0	1	0	0.5
0	-0.2	1	1	0	0	0.6

4.1.2 USPS dataset

The handwritten digits USPS dataset [23] is a digital dataset created by the United States by scanning envelopes automatically. Postal Service contains 9298 images divided into 7291 images for training and 2007 images for testing which is 80%:20% divisions. There are 256 features. The images here have been deslanted and the size is normalized, resulting in (16, 16) grayscale images.

4.2 Performance metrics

The used performance metrics are (1) accuracy, (2) precision, (3) recall, and (4) F1-score. They, and their equations, are represented in Table 14.

where TP (True Positive) is the number of samples classified positive correctly, TN (True Negative) is the number of samples classified negative correctly, FP (False Positive) is the number of samples wrongly classified positive, and FN (False Negative) is the number of samples wrongly classified negative.

Table 14 The used Performance Metrics

Metric	Formula	Description
Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$	The proportion of true results to the total number of samples
Precision	$\frac{TP}{TP+FP}$	The percentage of positive labeled samples that were predicted as positive
Recall	$\frac{TP}{TP+FN}$	The proportion of actual samples that are correctly identified
F1-Score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Accuracy of the model considering both precision and recall

Table 15 Filter phase results using different methods

CHI square	Sorting of the features from Top-1 to Top-5				
	Temperature	Season	Date	Events	Holiday
Mutual Information	Temperature	Season	Date	Holiday	Events
Feature Importance	Temperature	Season	Date	Holiday	Events
Most Correlated Features	Temperature	Holiday	Events	Season	Date
ACC2	Temperature	Events	Season	Holiday	Date
AACC2	Temperature	Events	Season	Date	Holiday
Proposed Method	Temperature	Date	Season	Events	Holiday

4.3 Evaluation of EUNITE

The current subsection shows a comparison between the proposed method and others filters methods (1) CHI square [7], (2) mutual information [7], (3) Feature Importance [56], (4) ACC2 [42], and (5) AACC2 [42]. The results are shown in Table 15.

Table 15 shows the top-ranked feature up to top-5. As shown, the top-1 is the same for all methods whereas the other top-k differs from one method to another. To prove that the proposed method accurately ranks the features, each method is tested separately during the second phase. Figures 8, 9, and 10 show the accuracy for each method using the Neural Network, Naïve Bayes, and Support vector machine respectively. The results of this comparison are shown in Table 16. where The accuracy in different feature selection techniques for all features with classifiers We find in the NN classifier given the highest accuracy 0.97 among other classification in NB classifier given 0.79 and in SVM classifier given 0.92, in Top 3 the NN classifier given the highest accuracy 0.97 among other classification in NB classifier given 0.82 and in SVM classifier given 0.95, and Top 4 the NN classifier given the highest accuracy 0.97 among other classification in NB classifier given 0.81 and in SVM classifier given 0.95, they present values close to Top 5, which shows the highest value using all the features and our

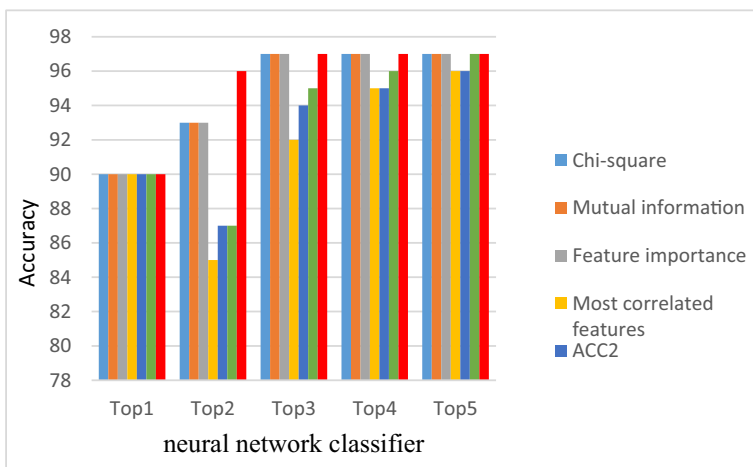


Fig. 8 Accuracy for different methods using Neural Network classifier

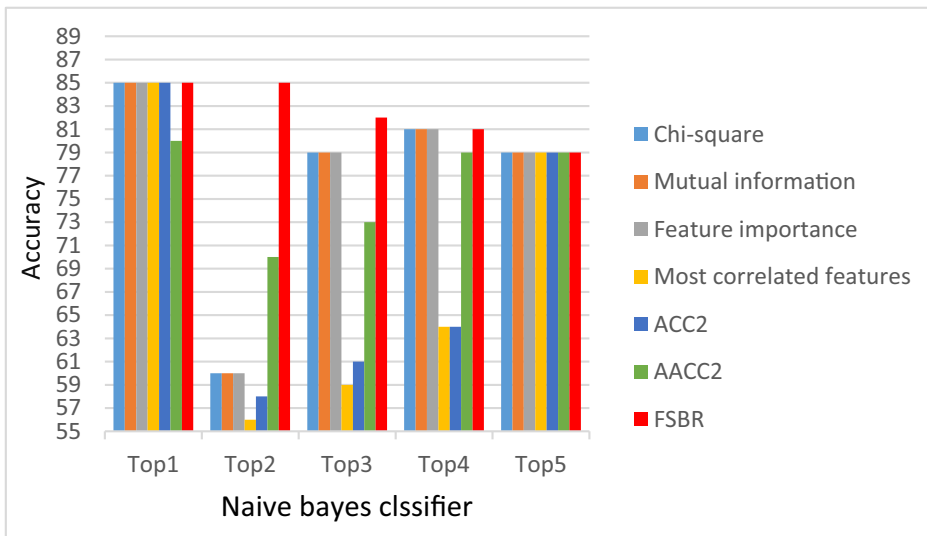


Fig. 9 Accuracy for different methods using Naïve Bayes classifier

proposed method is of the highest values. From the results and to take advantage of the values of the different classifiers in each method and get the best values from merging them. we can calculate the average to each top to the proposed filter method and the other different feature selection techniques for each Top with classifiers, In Fig. 11 illustrated the average result in each top.

From previous figures, we can conclude that the top 3, top 4, and top 5 ranked features are the best to use but for the sake of simplicity and less time and hardware computation, the top 3

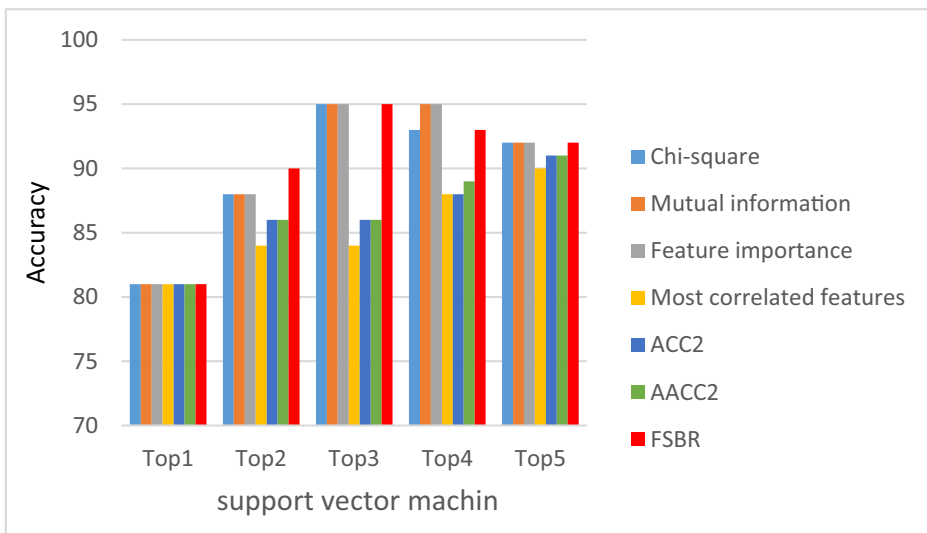


Fig. 10 Accuracy for different methods using SVM classifier

Table 16 The result of the FSBR and other methods with different classifiers on the EUNITE dataset

	Methods	NN (%)	NB (%)	SVM (%)	Average (%)
TOP1	Chi-square	90	85	81	85.33
	mutual information	90	85	81	85.33
	feature importance	90	85	81	85.33
	Most correlated features	90	85	81	85.33
	ACC2	90	85	81	85.33
	AACC2	90	80	81	83.7
	FSBR	90	85	81	85.33
	TOP2	Chi-square	93	60	88
mutual information		93	60	88	80.33
feature importance		93	60	88	80.33
Most correlated features		85	56	84	75
ACC2		87	58	86	77
AACC2		87	70	86	81
FSBR		96	85	90	90.33
TOP3		Chi-square	97	79	95
	mutual information	97	79	95	90.33
	feature importance	97	79	95	90.33
	Most correlated features	92	59	84	78.33
	ACC2	94	61	86	80.33
	AACC2	95	73	86	84.7
	FSBR	97	82	95	91.33
	TOP 4	Chi-square	97	81	93
mutual information		97	81	95	91
feature importance		97	81	95	91
Most correlated features		95	64	88	82.33
ACC2		95	64	88	82.33
AACC2		96	79	89	88
FSBR		97	81	93	90.33
TOP 5		Chi-square	97	79	92
	mutual information	97	79	92	89.33
	feature importance	97	79	92	89.33
	Most correlated features	96	79	90	88.33
	ACC2	96	79	91	88.7
	AACC2	97	79	91	89
	FSBR	97	79	92	89.33

is chosen to continue our comparisons between the proposed filter method and other different filters. The implementation of these different features selection techniques provides a different subset of the selected features. However, the Chi-square, mutual information, and feature importance techniques provide the same features subset. For this reason, Chi-square, mutual information, and feature importance have the same “Accuracy”, “Precision”, “Recall”, and “F1-score” values based on the average classifier. On the other hand, the other techniques which are; Most correlated features, ACC2, AACC2, and FSBR have different results of the measurements as shown in Table 17.

According to the obvious results and Figs. 12, 13, 14, 15, it is noticed that the proposed features selection method represented in FSBR demonstrates the “Accuracy”, “Precision”, “Recall”, and “F1-score”. Thus, It has been proven that FSBR is the most efficient methodology. On the other hand, it is found that Most correlated features introduce the worst performance in terms of “Accuracy”, “Precision”, “Recall”, and “F1-score”. For Most correlated features and FSBR, the error reaches 22% and 9% respectively. FSBR is better than Most correlated features because FSBR is based on using a hybrid technique that combines filter and

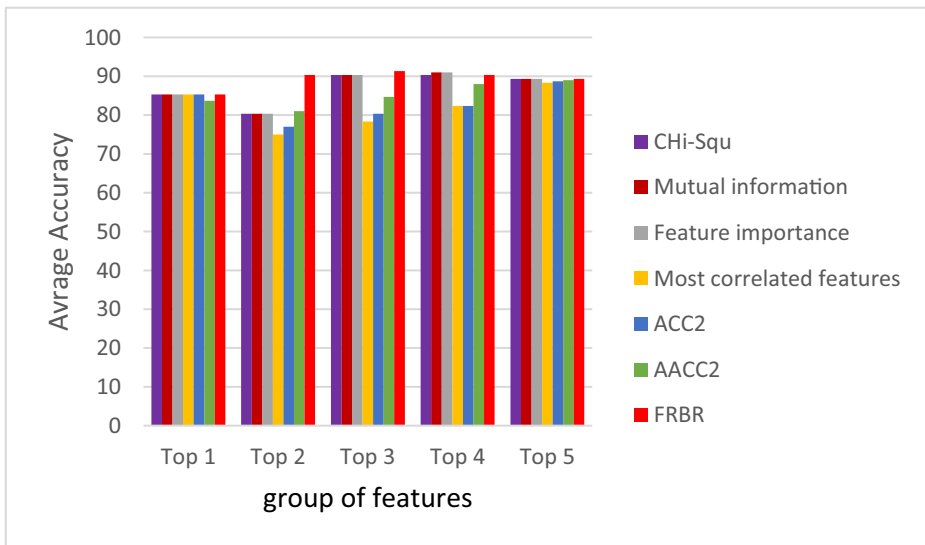


Fig. 11 The average Accuracy for the different methods

wrapper methods. Firstly, the hybrid technique composes of three filter methods called FRW, FE, and IG which are cooperated for providing the best rank of features by the fuzzy method. Finally, using NN, NB, and SVM to Combine them to get the best value is applied as a wrapper method to determine exactly the most significant subset of features. On the other hand, Most correlated features are based only on the filter selection method, so it provides a less accurate subset of features.

Finally, FSBR is better than other techniques to select the best subset of features that is improving the performance of the classifier or the prediction model using only three features.

4.4 Generalization using USPS

The USPS dataset is used to prove the applicability of our proposal on a big dataset. Table 18 shows a comparison between our proposed method for feature ranking and selection with different feature selection algorithms in [31]. Comparison criterion is made as follows, the whole number of features go through each feature selection algorithm including the proposed one then the top-ranked features are used to train a Neural Network, and finally, the test results

Table 17 The performance metrics of FSBR compared to other methods on the EUNITE dataset

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Chi-square	90	89	90	89
mutual information	90	89	90	89
feature importance	90	89	90	89
Most correlated features	78	79	78	77
ACC2	80	80	80	77
AACC2	84	84	83	80
FSBR	91	90	91	90

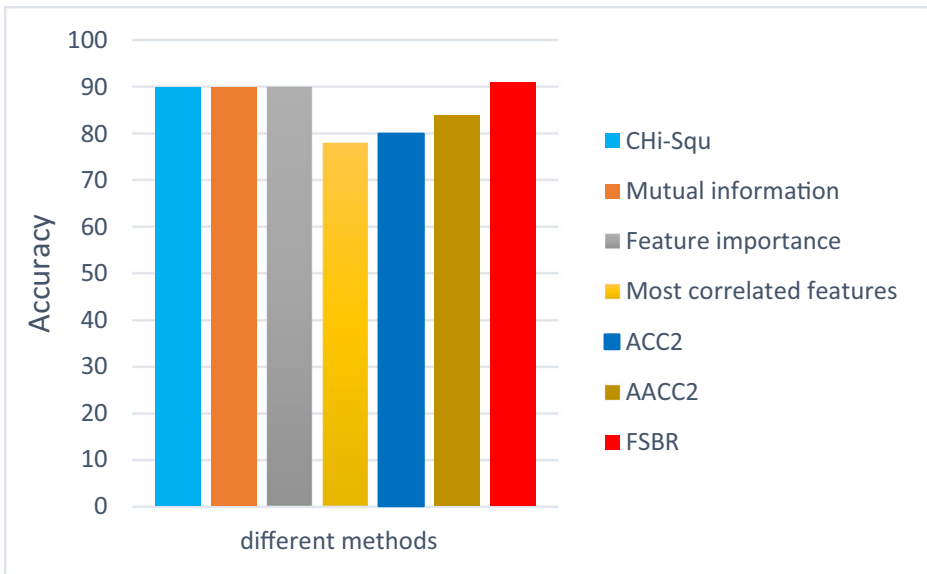


Fig. 12 Accuracy for Top 3 ranked features for different filter methods

are used to decide which method is the best. Figure 16 illustrates the result accuracy, precision, and recall of the proposed method and the different feature selection algorithms in Table 18. From that, our proposed method has reported the best result.

As shown in Table 18, the least amount of features to gain the maximum accuracy were 10 features, the number is the same for each method but each method chooses its features based

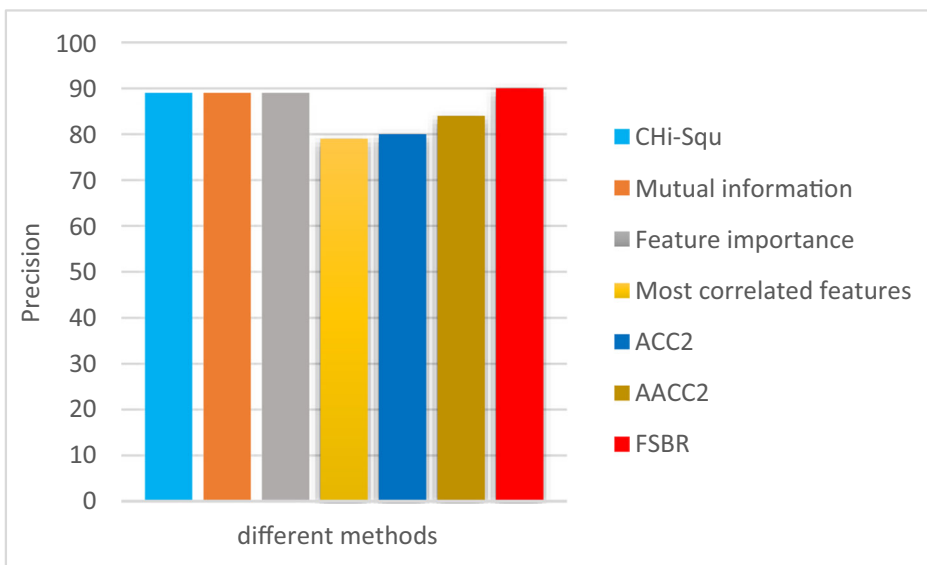


Fig. 13 Precision for Top 3 ranked features for different filter methods

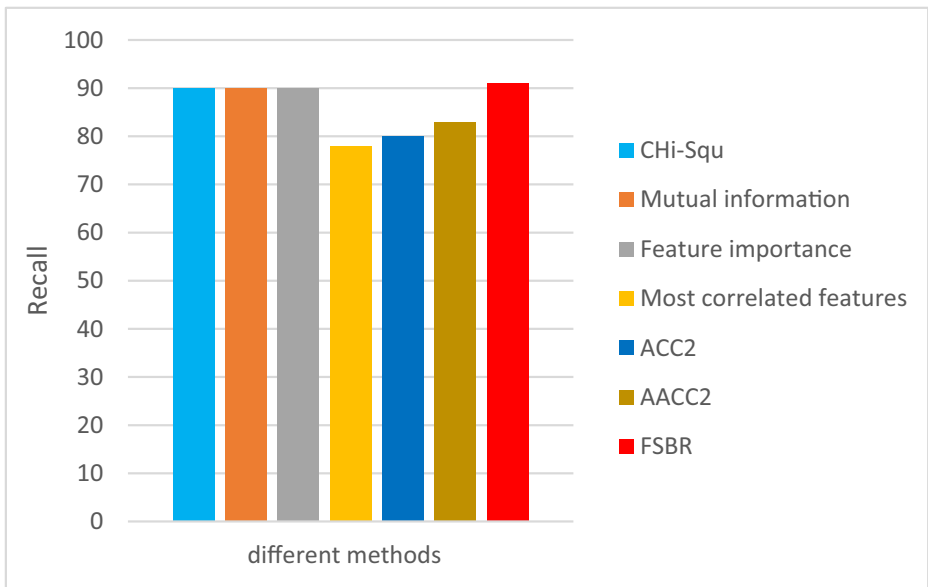


Fig. 14 Recall for Top 3 ranked features for different filter methods

on their ranks. The proposed method succeeded to get the highest accuracy compared to 12 methods and is very close to the others. Also, it got the highest precision among others. The recall was better than 10 methods.

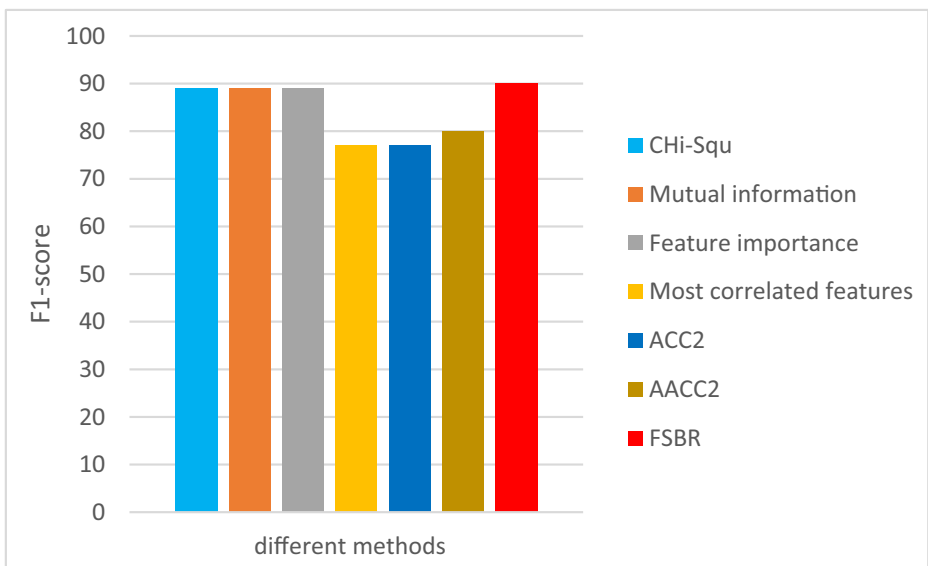


Fig. 15 F1-score for Top 3 ranked features for different filter methods

Table 18 The performance metrics of FSBR compared to other methods on the USPS dataset

Feature Selection Method	Number of Selected Features	Accuracy (%)	Precision (%)	Recall (%)
WCC [31]	10	85	86	85
LCA [31]	10	85	83	85
GA [31]	10	85	86	85
PSO [31]	10	87	88	80
ACO [31]	10	85	85	85
ICA [31]	10	86	89	86
LA [31]	10	89	89	89
HTS [31]	10	81	82	81
FOA [31]	10	83	84	83
DSOS [31]	10	82	80	82
CUK [31]	10	84	84	84
ReliefF [4]	64	88.64	60.06	95.88
2D ensemble [4]	57.6	90.12	63.56	95.85
Framework to deal with stability and FA [4]	57.6	90.12	63.56	95.85
Proposed Method	10	90	93	88

5 Conclusions and future work

As a recap, the proposed method is a feature selection method based on filter and wrapper techniques. The filter phase consists of three different filters: relative weight ranking, effectiveness ranking, and information gain ranking. In the wrapper phase, we used three different classifiers to select the least amount of top-ranked features (from the previous phase) without affecting the performance. The used were classifiers Neural Network, Naïve Bayes, and Support Vector Machine. Hence, our main contribution was to improve the smart electrical grid by optimizing the data being sent to fog and cloud. However, only important data is selected while any other repetitive and irrelevant data is dropped, maintaining the performance of the system. Therefore, we have proposed a new feature selection method that could successfully choose only important data with proving its correctness by applying it on two

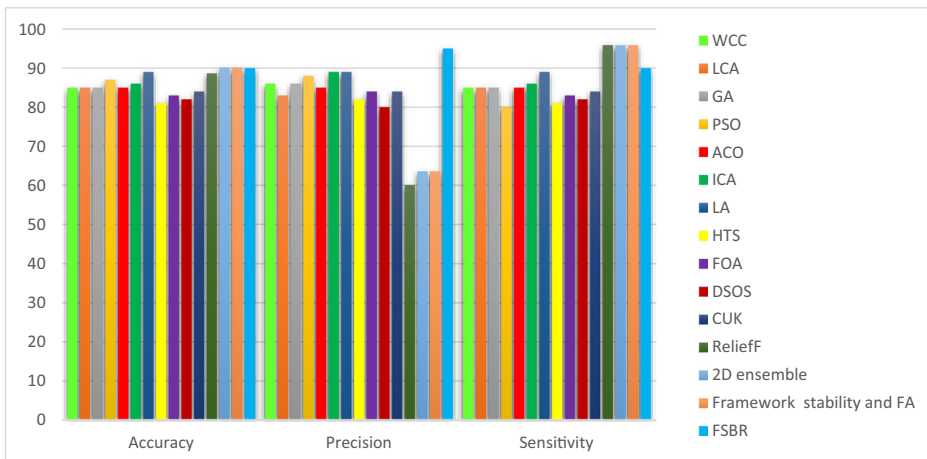


Fig. 16 The result to accuracy, precision, and sensitivity

different datasets. EUNITE, which is related to the electrical field, and USPS, which is not related to the electrical field but is about images. In both cases, results were satisfying enough to put our proposed method in a comparison with other methods. Experimental results have shown that the proposed feature selection technique provides more accurate results than the existing methods in terms of accuracy, precision, recall, and F1-score. FSBR provide accuracy, precision, recall, and F1-score values that reached 91%, 0.90%, 0.91%, and 0.90% respectively in EUNITE 90%, 93%, and 88% were the best accuracy, precision, and recall respectively use the USPS dataset. In future work, we will work on the second layer, the user load forecasting strategy, and the possibility of saving on electricity consumption by predicting the load per user in the short-term and long term.

Authorship We confirm that the manuscript has been read and approved by all named authors. We confirm that the order of authors listed in the manuscript has been approved by all named authors.

Authors contribution We the undersigned declare that this manuscript is original, has not been published before, and is not currently being considered for publication elsewhere. We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us. We understand that the Corresponding Author is the sole contact for the Editorial process. She is responsible for communicating with the other authors about progress, submissions of revisions, and final approval of proofs.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Declarations

Conflict of interest We wish to confirm that, there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Intellectual property We confirm that, we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, concerning intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abualigah L, Dulaimi AJ (2021) A novel feature selection method for data mining tasks using hybrid sine cosine algorithm and genetic algorithm. *Clust Comput*:1–16
2. Ahmad A, Javaid N, Alrajeh N, Khan Z, Qasim U, Khan A (2015) A modified feature selection and artificial neural network-based day-ahead load forecasting model for a smart grid. *Appl Sci* 5(4):1756–1772
3. Ahmed S, Lee Y, Hyun SH, Koo I (2018) Feature selection-based detection of covert cyber deception assaults in smart grid communications networks using machine learning. *IEEE Access* 6:27518–27529
4. Alhamidi MR, Jatmiko W (2020) Optimal feature aggregation and combination for two-dimensional ensemble feature selection. *Information* 11(1):38
5. Ali SH, et al (2020) A Gen-Fuzzy Based Strategy (GFBS) for Web Service Classification. *Wire Person Commun* 113(4):1917–1953
6. Al-Turjman F, Abujubbeh M (2019) IoT-enabled smart grid via SM: an overview. *Futur Gener Comput Syst* 96:579–590
7. Bahassine S, Madani A, al-Sarem M, Kissi M (2020) Feature selection using an improved chi-square for Arabic text classification. *Journal of King Saud University-Computer and Information Sciences* 32(2):225–2, 231
8. Bellavista P, Berrocal J, Corradi A, Das SK, Foschini L, Zanni A (2019) A survey on fog computing for the internet of things. *Pervasive and mobile computing* 52:71–99
9. Cilia ND, de Stefano C, Fontanella F, Scotto di Freca A (2019) A ranking-based feature selection approach for handwritten character recognition. *Pattern Recogn Lett* 121:77–86
10. da Costa NL et al (2021) Evaluation of feature selection methods based on artificial neural network weights. *Expert Syst Appl* 168:114312
11. Darwish A, Hassani AE, Elhoseny M, Sangaiah AK, Muhammad K (2019) The impact of the hybrid platform of internet of things and cloud computing on healthcare systems: opportunities, challenges, and open problems. *J Ambient Intell Humaniz Comput* 10(10):4151–4166
12. Dastjerdi AV, Buyya R (2016) Fog computing: helping the internet of things realize its potential. *Computer* 49(8):112–116
13. Deng X, Li Y, Weng J, Zhang J (2019) Feature selection for text classification: a review. *Multimed Tools Appl* 78(3):3797–3816
14. Dileep G (2020) A survey on smart grid technologies and applications. *Renew Energy* 146:2589–2625
15. European Network on Intelligent Technologies for Smart Adaptive Systems (n.d.) <https://www.eunite.org/>. The competition page is: <http://neuron.tuke.sk/competition/>. Accessed 27 June 2021
16. Gan J, Wen G, Yu H, Zheng W, Lei C (2020) Supervised feature selection by self-paced learning regression. *Pattern Recogn Lett* 132:30–37
17. Ghobaei-Arani M et al (2019) Resource management approaches in fog computing: a comprehensive review. *Journal of Grid Computing*:1–42
18. Ghosh P, Azam S, Jonkman M, Karim A, Shamrat FMJM, Ignatious E, Shultana S, Beeravolu AR, de Boer F (2021) Efficient prediction of cardiovascular disease using machine learning algorithms with relief and LASSO feature selection techniques. *IEEE Access* 9:19304–19326
19. Goulden M, Bedwell B, Rennick-Egglestone S, Rodden T, Spence A (2014) Smart grids, smart users? The role of the user in demand side management. *Energy Res Soc Sci* 2:21–29
20. Hafeez G, Alimgeer KS, Qazi AB, Khan I, Usman M, Khan FA, Wadud Z (2020) A hybrid approach for energy consumption forecasting with a new feature engineering and optimization framework in smart grid. *IEEE Access* 8:96210–96226
21. Han W, Feng R, Wang L, Cheng Y (2018) A semi-supervised generative framework with deep learning features for high-resolution remote sensing image scene classification. *ISPRS J Photogramm Remote Sens* 145:23–43
22. Hancer E (2019) Differential evolution for feature selection: a fuzzy wrapper-filter approach. *Soft Comput* 23(13):5233–5248
23. Handwritten Digits USPS dataset (n.d.) <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#usps>. Accessed on 5 August 2021.
24. Huang Y, Jin W, Yu Z, Li B (2020) Supervised feature selection through deep neural networks with pairwise connected structure. *Knowl-Based Syst* 204:106202
25. Javadzadeh G, Rahmani AM (2020) Fog computing applications in smart cities: a systematic survey. *Wirel Netw* 26(2):1433–1457
26. Kumar P, Lin Y, Bai G, Paverd A, Dong JS, Martin A (2019) Smart grid metering networks: a survey on security, privacy and open research issues. *IEEE Communications Surveys & Tutorials* 21(3):2886–2927
27. Lim H, Kim D-W (2021) Pairwise dependence-based unsupervised feature selection. *Pattern Recogn* 111: 107663

28. Liu H, Zhou MC, Liu Q (2019) An embedded feature selection method for imbalanced data classification. *IEEE/CAA Journal of Automatica Sinica* 6(3):703–715
29. Mafarja MM, Mirjalili S (2019) Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection. *Soft Comput* 23(15):6249–6265
30. Mafarja M, Aljarah I, Faris H, Hammouri AI, al-Zoubi A'M, Mirjalili S (2019) Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Syst Appl* 117:267–286
31. Masoudi-Sobhanzadeh Y, Motieghader H, Masoudi-Nejad A (2019) FeatureSelect: a software for feature selection based on machine learning approaches. *BMC bioinformatics* 20(1):170
32. Mekki K, Bajic E, Chaxel F, Meyer F (2019) A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT express* 5(1):1–7
33. Mukherjee M, Shu L, Wang D (2018) Survey of fog computing: fundamental, network applications, and research challenges. *IEEE Communications Surveys & Tutorials* 20(3):1826–1857
34. Neggaz N, Ewees AA, Elaziz MA, Mafarja M (2020) Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Syst Appl* 145:113103
35. Niu W-J et al (2021) Short-term electricity load time series prediction by machine learning model via feature selection and parameter optimization using hybrid cooperation search algorithm. *Environ Res Lett* 16(5):055032
36. Ozger M, Cetinkaya O, Akan OB (2018) Energy harvesting cognitive radio networking for IoT-enabled smart grid. *Mobile Networks and Applications* 23(4):956–966
37. Pande SK et al (2021) A resource management algorithm for virtual machine migration in vehicular cloud computing. *Computers, Materials & Continua* 67(2):2647–2663
38. Priyanka E et al (2021) Review analysis on cloud computing based smart grid technology in the oil pipeline sensor network system. *Petroleum Research* 6(1):77–90
39. Rabie AH, Ali SH, Ali HA, Saleh AI (2019) A fog based load forecasting strategy for smart grids using big electrical data. *Clust Comput* 22(1):241–270
40. Rai S, De M (2021) Analysis of classical and machine learning based short-term and mid-term load forecasting for smart grid. *Int J Sustain Energy* 40(9):821–839
41. Rehmani MH, Davy A, Jennings B, Assi C (2019) Software defined networks-based smart grid communication: a comprehensive survey. *IEEE Communications Surveys & Tutorials* 21(3):2637–2670
42. Šahin DÖ, Kılıç E (2019) Two new feature selection metrics for text classification. *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije* 60(2):162–171
43. Saleh AI, el Desouky AI, Ali SH (2015) Promoting the performance of vertical recommendation systems by applying new classification techniques. *Knowl-Based Syst* 75:192–223
44. Sayed GI, Hassanien AE, Azar AT (2019) Feature selection via a novel chaotic crow search algorithm. *Neural Comput & Applic* 31(1):171–188
45. Shaban WM, Rabie AH, Saleh AI, Abo-Elsoud MA (2020) A new COVID-19 patients detection strategy (CPDS) based on hybrid feature selection and enhanced KNN classifier. *Knowl-Based Syst* 205:106270
46. Shahzad K, Iqbal S, Mukhtar H (2021) Optimal fuzzy energy trading system in a fog-enabled smart grid. *Energies* 14(4):881
47. Singer G, et al (2020) A weighted information-gain measure for ordinal classification trees. *Expert Systems with Applications* 152:113375
48. Singh SP, Nayyar A, Kumar R, Sharma A (2019) Fog computing: from architecture to edge computing and big data processing. *J Supercomput* 75(4):2070–2105
49. Stiawan D et al (2020) CICIDS-2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access* 8:132911–132921
50. Tang B, Zhang L (2020) Local preserving logistic I-relief for semi-supervised feature selection. *Neurocomputing* 399:48–64
51. Tom RJ, Sankaranarayanan S, Rodrigues JJPC (2020) Agent negotiation in an IoT-fog based power distribution system for demand reduction. *Sustainable Energy Technologies and Assessments* 38:100653
52. Tushar W, Yuen C, Chai B, Huang S, Wood KL, Kerk SG, Yang Z (2016) Smart grid testbed for demand focused energy management in end user environments. *IEEE Wirel Commun* 23(6):70–80
53. Verma AK et al (2020) Skin disease prediction using ensemble methods and a new hybrid feature selection technique. *Iran Journal of Computer Science*:1–10
54. Wan Y, Ma A, Zhong Y, Hu X, Zhang L (2020) Multiobjective hyperspectral feature selection based on discrete sine cosine algorithm. *IEEE Trans Geosci Remote Sens* 58(5):3601–3618
55. Wang X, Guo B, Shen Y, Zhou C, Duan X (2019) Input feature selection method based on feature set equivalence and mutual information gain maximization. *IEEE Access* 7:151525–151538
56. Wei G, Zhao J, Feng Y, He A, Yu J (2020) A novel hybrid feature selection method based on dynamic feature importance. *Appl Soft Comput* 93:106337

57. Yoldaş Y, Önen A, Muyeen SM, Vasilakos AV, Alan İ (2017) Enhancing smart grid with microgrids: challenges and opportunities. *Renew Sust Energ Rev* 72:205–214
58. Yousaf A, Asif RM, Shakir M, Rehman AU, S. Adrees M (2021) An improved residential electricity load forecasting using a machine-learning-based feature selection approach and a proposed integration strategy. *Sustainability* 13(11):6199
59. Zhang Z, Hong W-C (2021) Application of variational mode decomposition and chaotic grey wolf optimizer with support vector regression for forecasting electric loads. *Knowl-Based Syst* 228:107297
60. Zhang X, Zhang Q, Chen M, Sun Y, Qin X, Li H (2018) A two-stage feature selection and intelligent fault diagnosis method for rotating machinery using hybrid filter and wrapper method. *Neurocomputing* 275: 2426–2439
61. Zhou P, Chen J, Fan M, du L, Shen YD, Li X (2020) Unsupervised feature selection for balanced clustering. *Knowl-Based Syst* 193:105417
62. Zhou H, Zhang JW, Zhou YQ, Guo XJ, Ma YM (2021) A feature selection algorithm of decision tree based on feature weight. *Expert Syst Appl* 164:113842
63. Zhu Y, Zhang X, Hu R, Wen G (2018) Adaptive structure learning for low-rank supervised feature selection. *Pattern Recogn Lett* 109:89–96

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