IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 July 2022, accepted 31 August 2022, date of publication 12 September 2022, date of current version 26 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3205618

SURVEY

A Comprehensive Survey on the Process, Methods, Evaluation, and Challenges of Feature Selection

MD RASHEDUL ISLAM[1], (Member, IEEE), AKLIMA AKTER LIMA[2], SUJOY CHANDRA DAS[2], M. F. MRIDHA[2], (Senior Member, IEEE), AKIBUR RAHMAN PRODEEP[2], AND YUTAKA WATANOBE[3], (Member, IEEE)

[1]Department of Computer Science and Engineering, University of Asia Pacific, Dhaka 1205, Bangladesh
[2]Department of Computer Science and Engineering, Bangladesh University of Business and Technology, Dhaka 1216, Bangladesh
[3]Department of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Japan

Corresponding author: M. F. Mridha (firoz@bubt.edu.bd)

ABSTRACT Feature selection is employed to reduce the feature dimensions and computational complexity by eliminating irrelevant and redundant features. A vast amount of increasing data and its processing generates many feature sets, which are reduced by the feature selection process to improve the performance in all types of classification, regression, clustering models. This study performs a detailed analysis of motivation and concentrates on the fundamental architecture of feature selection. This study aims to establish a structured formation related to popular methods such as filters, wrappers and, embedded into search strategies, evaluation criteria, and learning methods. Different methods organize a comparison of the benefits and drawbacks followed by multiple classification algorithms and standard validation measures. The diversity of applications in multiple domains such as data retrieval, prediction analysis, and medical, intrusion, and industrial applications is efficiently highlighted. This study focuses on some additional feature selection methods for handling big data. Nonetheless, new challenges have surfaced in the analysis of such data, which were also addressed in this study. Reflecting on commonly encountered challenges and clarifying how to obtain the absolute feature selection method are the significant components of this study.

INDEX TERMS Feature selection, dimension reduction, optimization, search strategy, evaluation criteria, learning methods, data mining, machine learning.

I. INTRODUCTION

Humans has become increasingly dependent on electronic devices such as mobile phones, and computers. Thus, the use of real-world applications is increasing, which include vast amounts of data with high dimensions. This dimensionality is responsible for making data analysis a time-consuming and challenging task. To solve this problem and handle datasets with noisy and redundant data, various dimension reduction techniques are used. *Dimensionality reduction* is used in the preprocessing phase to address feature reduction problems. The goal of the feature reduction challenge is to reduce the size of the original datasets while maintaining accuracy. The most commonly used dimensionality reduction processes are *Feature extraction (FE)* and *feature selection (FS)*. A feature is a unique quantified characteristic of the observation process. Not all features are required to extract relevant information from datasets. Several features may be redundant or irrelevant for various machine learning, deep learning and data science approaches. Some may mislead clustering results, thus decreasing the quality of the model. Throughout this instance, selecting a subset of the original features will almost always result in improved performance. Feature selection algorithms in supervised learning optimize some function of predicted accuracy. Unsupervised learning, on the other hand lacks class labels, and runs the risk of retaining all or only a subset of significant attributes. Limiting the number of features also improves readability. It relieves

The associate editor coordinating the review of this manuscript and approving it for publication was Jon Atli Benediktsson.

the problem that specific unsupervised learning algorithms under-perform when dealing with high-dimensional data.

In Figure 1, the upper portion of feature space depicts an example of a non-essential or irrelevant feature. It's important to remember that feature dimension 2 seems to have no consequence on cluster discrimination. When used independently, feature dimension 2 produced an unremarkable single cluster structure. It is worth noting that irrelevant features can skew clustering results. The concept of feature redundancy is illustrated in the lower portion of Figure 1. It is important to remember that the data can be sorted in the same way using only the feature dimensions of 1 or 2. As a result, feature dimensions 1 and 2 are believed to be redundant.

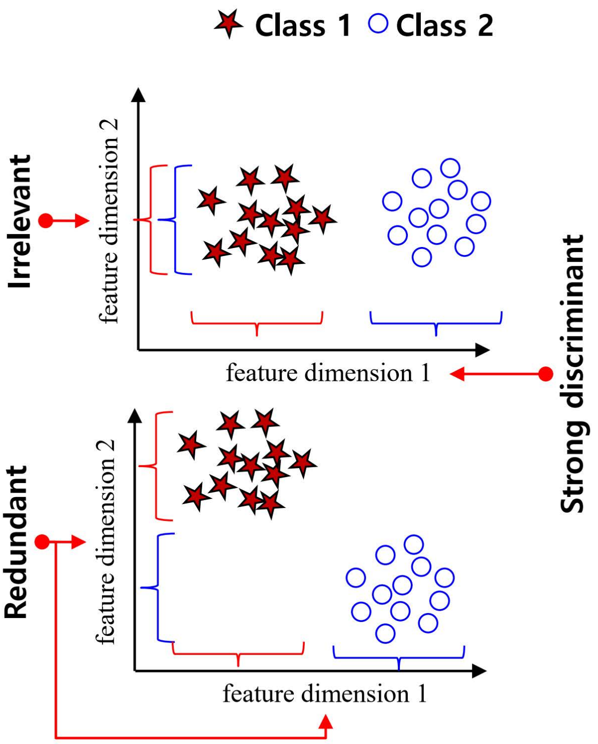


FIGURE 1. The figure shows how the feature dimensions select the redundant and irrelevant features from different classes.

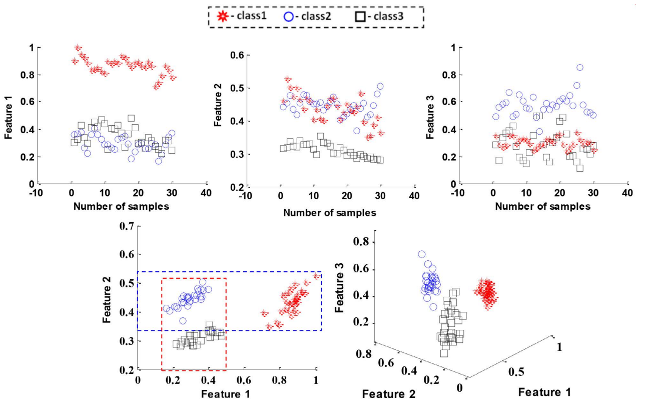


FIGURE 2. The figure illustrates the comparison between several classes within two dimensional and three dimensional feature spaces.

With adequate information about irrelevant and relevant features, dimensionality reduction can be achieved. In Figure 2, three different classes are indicated in different shapes and colors (red, black, and blue). In the above part of Figure 2, Feature 1, set on a one dimensional space, shows that classes 2 and classes 3 overlap whether classes 1 is separable. Feature 2, same as Feature 1, shows that classes 1 and 2 overlap. Finally, in Feature 3, the three classes overlap with each other and are merely inseparable. The lower-left section of Figure 2 depicts a combination of Feature 1 and Feature 2. In this combination, classes 1 and 2 overlap in terms of Feature 2. From the standpoint of Feature 1, it is debatable whether Class 1 remains distinct from Classes 2 and 3. The lower-left section of Figure 2 shows the combination of features 1 and 2. In terms of Feature 2, classes 1 and 2 overlap in this combination. Whether Class 1 remains separable from Classes 2 and 3 from the perspective of Feature 1 is debatable. In two-dimensional space, the combination reveals that each of the three classes is easily distinguishable. Combining these three features allows for easier differentiation of classes in the three-dimensional space depicted in the lower-left section of Figure 2. However, three-dimensional space is not required as in two-dimensional space, all three classes are separable if one dimension space is insufficient. Using two features rather than three is an example of both dimensionality reduction, and feature selection. Furthermore, the motivations and goals of feature selection were purposefully made more visible. An expeditious review of feature selection's goal points to reduce computational complexity and, as a result, improve system performance parameters such as accuracy. It also aims to reduce large dimensionality, in which some dimensions of some instances interfere with each other and affect the performance. It also aims to extract meaningful rules from the classifier and remove redundant features to reduce complexity.

Furthermore, in some cases, these feature reduction challenges or activities can be named classification, clustering, regression of data, and search strategy. These activities have been developed formed very recently with an increasing number of studies of feature selection. However, these activities or challenges started with a regression problem that identifies the formation of the FS history. In 1924 R. A. Fisher introduced a trial of variable selection for regression while discussing an article [1] presented by A. J. Miller to the Royal Statistical Society. Later in the 1940s, with limited computing power available, the trial faced some advancements. A study on the rationale for variable selection by Hotelling [2] illuminated previous approaches to solve this problem. Advancements in computing power in the early 1960s provided significant impetus for research in this area. The majority of early research was conducted by statisticians and focused on linear regression, such as Hocking [3], who conducted a literature review on variable selection for linear regression. Variable selection research has expanded to include classification and clustering issues as well. This growth has fascinated a wide range of artificial intelligence, machine learning, and data mining experts. As a result, phrase *variable selection* is gradually being phased out in favor of *feature selection*.

With the evolution of time, feature selection has become more structured and usable. This structure offers the basic architecture of the FS. The FS process is divided into four stages: subset generation or search, subset evaluation, stopping criterion, and result validation. Figure 3 depicts how the original set of documents searches for a relevant subset and then evaluates the subset to determine its quality. The evaluated subset must perform the stopping criterion step, which is known as over-fitting removal. In the validation phase, the process results in a relevant subset of features.

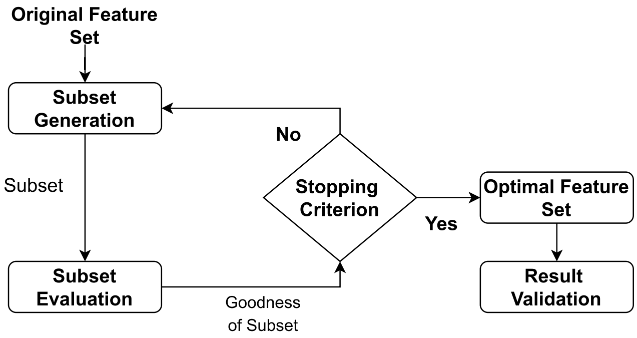


FIGURE 3. The basic architecture of Feature subset Selection shows how the feature subsets are selected from the original set of features.

- **Subset generation or search:** The original sets of documents go through the subset generation phase. In this phase, each state provides a candidate subset for the evaluation [4]. However, two concerns dictate the nature of the subset generation process- successor generation determines the search beginning point, which affects the search direction. A search can begin with an empty set, entire set, or a completely random subset [5]. Forward, backward, compound, weighted, and random approaches can be used to determine the search beginning points for each state. Search organization is responsible for selecting features using a specific strategy, such as, sequential, exponential, or random searches.
- **Subset Evaluation:** The candidate feature subsets must be examined using specific criteria to find the optimal feature subset based on the goodness measure. Additional evaluation criteria might not agree with an optimal feature subset determine through one measure. There are two widely used evaluation criteria based on the algorithms' reliance and independence [4]. One type of criterion is the criterion that is independent of one another and is commonly used in filter algorithms. It focuses on measuring the fundamental features of a dataset without using a data mining technique. The most common criteria for determining dependence are the probability of errors and information measurements. Another type of criterion that wrapper models use is the criterion that is dependent on one another. A unique mining algorithm was used to determine the criterion. The performance of the mining algorithm performance determines the quality of the feature subset. For a predefined mining algorithm, the dependent criterion typically outperforms an independent criterion. However, the selected feature subset may not be suitable for other mining techniques, and the computational cost is high. Unidentified instance forecasting accuracy is commonly used to identify a feature subset that yields high testing accuracy for classification problems [6].
- **Stopping criteria:** After the previous phase, the FS process requires a stopping criterion [4]. A suitable stopping criterion reduces the time it takes to locate the best feature subset and eliminates over-fitting. The decisions made in the preceding steps influence the selection of the stopping criterion. The following are among the most regularly used stopping criteria-
  - -- Based on the evaluation function.
  - -- Predetermined number of features.
  - -- Number of iterations and the proportion of two successive iteration steps.
- **Optimal feature set:** A subset of a specified feature set is the optimal feature set. The optimal subset minimizes a user-defined cost function (information-or performance-related, depending on the application). The optimal feature set reduces the number of inadvertently selected features by half while maintaining constant true positive rates. It is more efficient in selecting appropriate variables, resulting in a model that is more straightforward, understandable, and accurate.
- **Result validation:** The results must be ambiguously validated. Experimenting with the entire feature set, rather than just a subset, is a common strategy. To validate the results, the efficiencies of the before-and-after feature selection trials were compared. Cross-validation [7], [8], Confusion matrix [9], Jaccard similarity-based measure [10], Rand Index [11], and other validation methods have been widely used.

Different subset evaluation and subset search techniques are floating around numerous research points in the taxonomy of FS. Furthermore, data mining and machine learning tasks such as classification, regression, clustering, and association necessitate the use of FS methods. Among these tasks, feature selection improves readability and interpretability. Based on these scenarios, FS can be divided into three categories based on its criteria, as depicted in Figure 4: search strategy, evaluation criteria, and learning methods. Furthermore, FS can be divided into evaluation criteria, search strategies, and learning methods.

Apart from a broad discussion on classification and taxonomy, this study presents some challenges in the field of feature selection. Furthermore, this study shows different aspects and the usefulness of feature selection. The contributions of this study are described as follows:

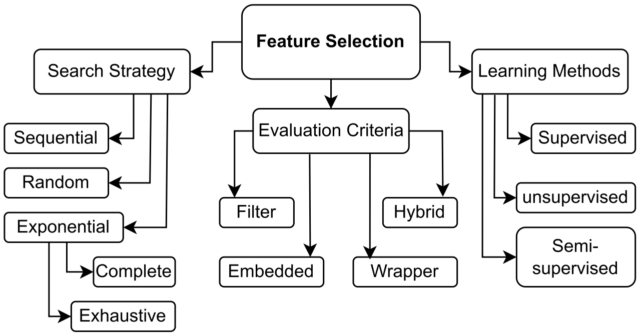


FIGURE 4. A taxonomy of feature selection shows the three criteria by which one can select relevant features.

- This paper precisely focuses on a standard architecture of feature selection that regulates the study's flow.
- The paper illustrates a broader taxonomy of feature selection and elaborates on various search, evaluation, and learning criteria.
- The paper shows several primarily used result validation and performance measure techniques as well.
- This paper then demonstrates the application of feature selection and classifies them into known sectors.
- In addition, this paper investigates the challenges of FS by applying the methods and exploring them to find a better way to handle them.

The rest of the paper is organized as follows: Section II shows the broad discussion of search categories from the taxonomy of FS. Section III focuses on methods that are classified based on evaluation criteria. A subset classification based on various learning methods is presented in Section IV. Different performance measurement and result validation techniques are described in Section V. Section VI presents a feature selection analysis for big data. Section VII presents the popular applications of FS. Some common challenges encountered during the evolution of FS are mentioned in Section VIII. Finally, Section IX brings the paper to a close.

II. FEATURE SUBSET SEARCH CATEGORIZATION

A search strategy aims to discover a feature subset containing 2^n where n is the number of features that maximizes the measurement function in the feature subset space [12]. Before starting the search strategy, there is a requirement to determine the search direction and starting point. There are several search directions: forward, backward, compound, weighted, and random approaches [4].

- **Forward:** Forward search is a phenomenon in which the search process begins with an empty set. New features were added recursively in each iteration.
- **Backward:** The backward elimination search begins with a complete set of features. It removes them individually until the required set of features is obtained.
- **Compound:** Compound search is a hybrid of forward and backward search mechanisms. Performing forward or backward steps based on corresponding values is an intriguing method. This permits novel interactions between features to be discovered.
- **Weighted:** The search space in weighted operators is a continuous process. All features were present in the solution to some extent. The successor has a different weight than that of the parent state. This is typically accomplished by selecting the available set of iterative instances.
- **Random:** The feature subset is constructed through a random search process, which involves repeatedly adding and removing features.

A search strategy can be implemented when the search direction is determined. Figure 4 depicts several search strategies that can be classified into three categories: Exponential algorithms [13], Sequential algorithms [14], and Randomized algorithms [15].

A. EXPONENTIAL ALGORITHM

Exponential algorithms evaluate a number of subsets that grow exponentially with the dimensionality of the search space also known as complete search. The most widely utilized and representative algorithms in this category are discussed below -

1) EXHAUSTIVE ALGORITHMS

Exhaustive searches are NP-hard [16], and sub-optimal methods such as forward selection [17] start small and make additions to improve performance. The other method is backward selection [18], which starts with all features and removes them to improve performance and is frequently utilized. An exhaustive search, such as the forward selection method, begins by obtaining the best one-component subset of the input features. It continues to search for the best two-component feature subset that can be composed of any combinations of input features. It is also the greedy-algorithm because it tries every possible feature combination and chooses the best. Figure 5 illustrates the exhaustive search.

2) COMPLETE SEARCH

A complete search is a strategy to find a solution to a problem by traversing the entire search space. It ensures that an optimal result is obtained based on the evaluation criteria employed. The exhaustive search part of the exponential search was regarded as complete. The fact that a search is complete does not imply that it is exhaustive. Various heuristic functions can be used to narrow the search space without decreasing the probability of obtaining the best solution. Consequently, even though the order of the search space is $O(2^N)$, fewer subsets are explored [19]. Two examples are branch and bind [13] and beam search [20].

- **Branch and Bound (BnB):** Branch and bound (BnB) solves discrete and combinatorial optimization issues and mathematical optimization problems [21]. The algorithm investigates the components of the tree, that are subsets of the optimal solution. It is applied to determine

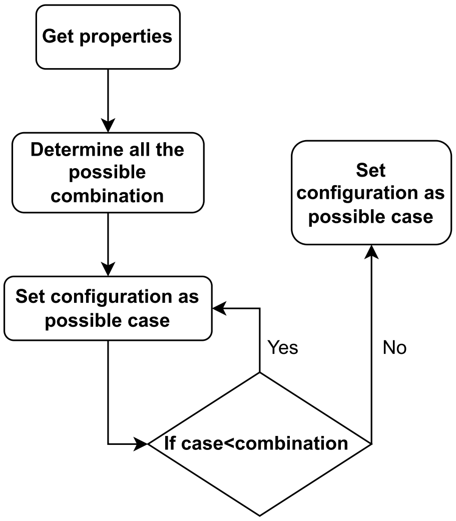

Get properties
Determine all the possible combination
Set configuration as possible case
Set configuration as possible case
Yes
No
If case<combination

FIGURE 5. This flowchart illustrates exhaustive algorithm.

the best solution for combinatorial, discrete, and fundamental mathematical optimization problems. Given an NP-Hard problem, a branch and bound method investigates the search space of possible solutions and determines the best solution [22]. Several studies [23], [24], [25], [26] used the BnB algorithm in their works.

- **Beam Search:** Beam search is a heuristic search strategy that expands the most intriguing node in a restricted collection to explore a graph. It utilizes the optimization of the breadth-first search [27].

B. SEQUENTIAL ALGORITHM

Sequential algorithms are employed to add or remove features sequentially. This algorithm tends to be trapped in local minima. Several sequential algorithms which have been utilized for decades. Some of these issues are discussed in the following sections.

1) SEQUENTIAL FORWARD SELECTION (SFS)

Sequential forward selection (SFS) is a technique in which features are sequentially assigned to empty candidates until the criterion is not altered [28]. Sequential feature selection techniques used to minimize an initial dimensional feature space to another dimensional feature subspace are included in a group of greedy search algorithms. The goal is to select a subset of features that are most relevant to the purpose, resulting in optimal computational performance while reducing overfitting by removing irrelevant information. The SFS performs best when the optimal subset has a small number of features. SFS has been utilized in some of the articles [29], [30], [31], [32].

2) SEQUENTIAL BACKWARD SELECTION (SBS)

The sequential backward selection approach intends to reduce the dimensionality of the initial feature subspace from N to K features with a minimum reduction in system performance [33]. This improves computational efficiency and reduces overfitting. The main goal is to eliminate features from the provided feature list of N features one by one until they reach the list of K-features. At each stage of the process, the feature that caused the least performance loss was removed. The feature approach is based on a combinatorial search method, in which a subset of features from a combination is chosen.The score for the subset was calculated and compared with other subsets. Several studies have been conducted using the SBS algorithm [32], [34], [35], [36].

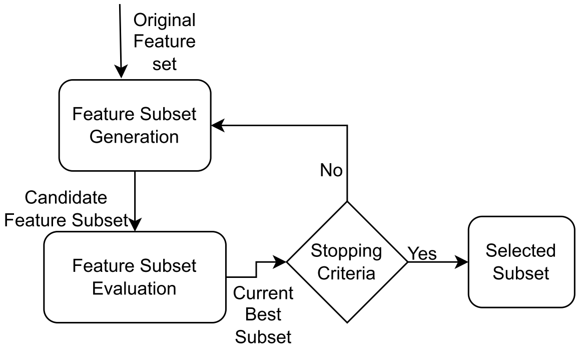


FIGURE 6. The graph shows how the thick lines in the search space identified by Sequential Forward Searching narrow as the algorithm approach the whole feature set.

3) SEQUENTIAL FORWARD FLOATING SELECTION (SFFS)

The Sequential Forward Floating Selection (SFFS) process involves counting backward steps after each forward step [37]. This process repeats the steps as long as the generated subsets are preferable to those initially considered at that level. Consequently, if the intermediate result at the actual level (of the relevant dimensions) cannot be increased, there are no backward steps. The same is true for the reverse counterpart of the procedure. Both algorithms support 'self-controlled backtracking,' allowing them to obtain effective results by dynamically altering the trade-off between the forward and backward steps. In this manner, they evaluated what they required without using any parameters [38]. Recently, the SFFS algorithm has been proposed [32], [39], [40], [41].

4) SEQUENTIAL BACKWARD FLOATING SELECTION (SBFS)

The whole set is used to begin the sequential floating backward selection (SFBS). As long as the objective function advances, SFBS takes forward steps after each backward step [32], [33], [41].

5) PLUS-L MINUS-R SELECTION (L MINUS R)

LRS (Plus-L Minus-R Selection) is a combination of SFS and SBS [42], [43]. The algorithm has two versions: one that starts with an empty set, and adds L features in each round before eliminating R features until the metric evaluation value is optimal. Conversely, the algorithm starts with the universal set, eliminates R features in each round, and then adds L features to achieve the best value for the evaluation metric. The selection of L and R is crucial in this algorithm. This algorithm has been utilized in some studies like [44], [45], [46].

6) BIDIRECTIONAL SEARCH (BDS)

Bidirectional Search (BDS) substitutes a single search graph with two smaller subgraphs, starting from the beginning and the destination vertices. In addition, the search closes whenever two graphs intersect. BDS employs both SFS and SBS in feature selection and terminates searching when both locate the same feature subset [47], [48].

C. RANDOM SEARCH ALGORITHM

Random search algorithms were employed to escape the local minima. These algorithms are known as heuristic search algorithms. It incorporates randomness into its search process. Several random search algorithms have been introduced over the years, some of which are discussed in the following section.

1) METAHEURISTIC ALGORITHMS

Optimization methods that aim to find the optimal (or near-optimal) solution to an optimization problem are called metaheuristic algorithms. These algorithms are derivative-free methods that are simple, flexible, and capable of avoiding local optima [49]. Metaheuristic algorithms exhibit stochastic behavior, as they begin their optimization process by producing random solutions. Unlike gradient search approaches, it does not require the derivative of the search space to be calculated. Metaheuristic algorithms are versatile and straightforward owing to their simple principle and easy implementation. The notable feature of metaheuristic algorithms is the extraordinary ability to prevent algorithms from converging prematurely. A flowchart of metaheuristic algorithm is illustrated in Figure 7. Metaheuristic algorithms can be classified into four types [50]:

1) Evaluation based: It is based on natural evolution, and begins with a population of randomly produced solutions. The best solutions are combined into these algorithms to produce new persons. Mutation, crossover, and optimum solution are used to create new individuals. Differential search [51], Stochastic fractal search algorithm [52], Backtracking search [53], and Synergistic fibroblast optimization [54] are examples of evaluation based algorithms.
2) Swarm intelligence-based algorithms: These algorithms are based on the social behaviors of insects, animals, fish, and birds. Particle Swarm Optimization (PSO), invented by Kennedy and Eberhart [55], is a prominent approach. It is based on the behavior of a flock of birds that fly across the search space to find an ideal site for them (position).
3) Physics based algorithms: The rules of physics were inspired by these algorithms. Search based algorithms such as, Gravitational search [56], Charged system search [57], Galaxy based search [58], Optimization algorithm such as, Electro magnetism [59], Spiral [60], Curved space [61], Ray [62], Gases Brownian Motion [63], Kinetic gas molecule [64], Colliding bodies [65], Water vaporization [66], Thermal exchange optimization [67] are some example of physics based algorithms. In addition, the black hole algorithm [68], Water cycle [69], Mind blast algorithm [70], Sine cosine algorithm [71], and Electro search algorithm [72] are physics based algorithms.
4) These methods are based on human behavior. Every person has a unique way of carrying out activities, that influences their overall success. League championship [73], Exchange market algorithm [74], Social emotion [75], Brain storm optimization [76], Jaya algorithm [77], Gaining sharing knowledge based algorithm [50] are examples of human behavior algorithms.

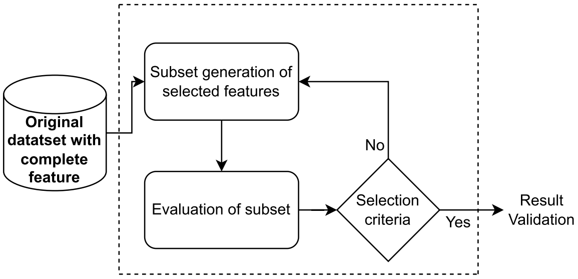

Original datatset with complete feature
Subset generation of selected features
Evaluation of subset
Selection criteria
No
Yes
Result Validation

FIGURE 7. This flowchart illustrates metaheuristic algorithm.

2) RANDOM GENERATION PLUS SEQUENTIAL SELECTION (RGSS)

Random search algorithms produce a subset of features at random and then apply additional algorithms to that subset [78]. Random generation plus sequential selection (RGSS) performs SFS and SBS on a randomly chosen subset of features to break free from the local optimum. Random search methods, however, rely on random parameters, making it difficult to replicate the experimental results [79]. This study utilized RGSS search algorithms [80], [81].

3) SIMULATED ANNEALING

A set of features was selected randomly to begin the simulated annealing process [82]. It is also possible to specify the number of iterations and obtain the model's prediction performance [83]. The existing feature set is then randomly included or excluded from a small fraction (1-5) of the features, and predicted performance of the new batch is determined. If the new features increase efficiency, the new set of features is maintained. If the new feature set under-performs, the acceptance probability is calculated using the equation for higher performance with greater values. The likelihood is a function of time and performance change, as well as a parameter c that controls how quickly the features are perturbed. Following the calculation of acceptance probability, a random uniform value was generated. If the initial feature set was used when the random value was greater than the acceptance probability, the new feature set was rejected and preserved. Simulated annealing can be helpful as it avoids local optimums in its search for the global optimum owing to the supply of randomness. It allows movements to state the error rates on a probabilistic basis illustrated in Figure 8. The recent utilized simulated annealing studies were [84], [85], [86], [87].

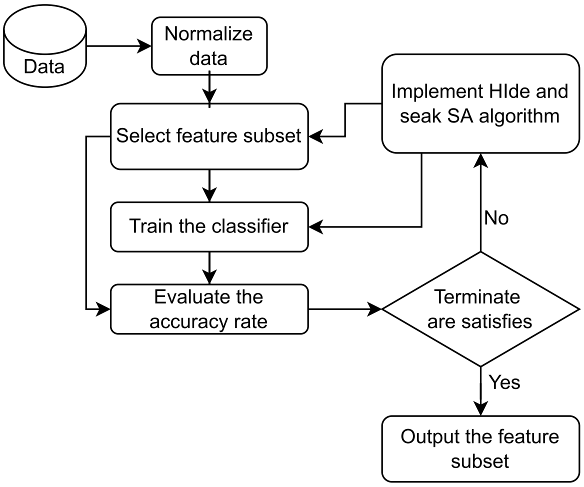

Data
Normalize data
Implement HIde and seak SA algorithm
Select feature subset
Train the classifier
No
Evaluate the accuracy rate
Terminate are satisfies
Yes
Output the feature subset

FIGURE 8. The graph illustrates how simulated annealing avoided local minima by allowing movements to develop state error rates on a probabilistic basis.

4) RANDOM HILL CLIMBING

Hill climbing is a type of heuristic search used to solve problems involving mathematical optimization [88]. It uses a set of inputs and suitable heuristic function. It aims to provide a decent solution to the problem. This search algorithm may not find the optimal solution; however, it employs a greedy strategy. At any position in the state space, the search continues only in the direction that optimizes the cost of the function in the hopes of eventually discovering the best answer [89]. The study successfully utilized the random hill climbing algorithm [90], [91], [92], [93].

5) MEMETIC ALGORITHM (MA)

An extension of the standard genetic algorithm is a memetic algorithm (MA). To minimize the chances of premature convergence, it employs a local search strategy. The crossover operator is a crucial component of the MA operation. The significant similarity between highly suited strings can guide a search [94]. Memetic algorithms are rapidly growing in the field of evolutionary computation studies [95], [96], [97].

6) LAS VEGAS ALGORITHM (LV's)

The Las Vegas algorithm make probabilistic decisions to assist in obtaining the correct answer quickly [98]. Randomness is used by one type of Las Vegas algorithm to lead their search so that a correct answer is ensured even if poor choices are made. Heuristic search methods are vulnerable to datasets with high order correlations. The LV's approach mitigates this concern by balancing the time spent on different cases [99].

7) DIFFERENTIAL EVOLUTION (DE)

Differential evolution (DE) is an evolutionary approach for generating real-valued multi-modal functions that are powerful and easy [100]. This is a population-based metaheuristic algorithm that iteratively improves a proposed solution through an evolutionary process. The parameters of the procedure are stored as floating-point variables that change when an essential mathematical operation is performed. During the mutation process, the modified most exemplary parameter values are merged into actual population vectors via a variable-length for each crossover procedure. These algorithms make few assumptions regarding the underlying optimization problem and can quickly explore enormous design spaces. The primary feature of the standard DE is that it has three control parameters that must be adjusted. The sample vector generation scheme and control parameter selection significantly impact effectiveness of DE in a specific optimization task [101]. To achieve good optimization results, trial vector generation strategy is selected and the system parameters for the optimization process is optimized. Choosing an appropriate control parameter is not always easy, and it can be time-consuming and difficult, especially for implementation. A flowchart illustrating differential evolution is shown in Figure 9.

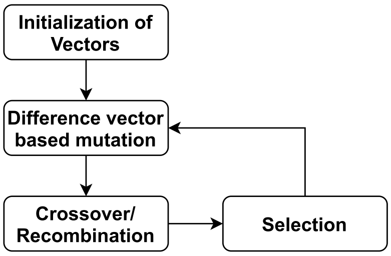


FIGURE 9. This flowchart illustrates differential evolution.

8) PARTICLE SWARM OPTIMIZATION (PSO)

Particle swarm optimization (PSO) algorithm is a metaheuristic algorithm founded on the principle of swarm intelligence capacity to resolve complicated mathematical problems in engineering [102], [103]. It is a computerized method for optimizing a challenge by constantly attempting to enhance a candidate solution for a particular quality measure [104]. A population (or swarm) of the initial solutions is used in the PSO method (particles). With a quick convergence time, a PSO may execute a global search across the entire search space. The movement of particles is determined by their well-known position in space and the orientation of the entire swarm. This enables real-time modification of the inertia weight, acceleration coefficients, and other computational factors, thereby increasing the effectiveness of the search. The PSO algorithm is notable for its simple concept, straightforward implementation, robustness with control parameters, and high computational efficiency [105].

9) GENETIC ALGORITHM (GA)

A genetic algorithm (GA) is a heuristic search strategy used to solve challenges involving search and optimization [106]. This is a strategy for dealing with restricted and unrestricted optimization problems that rely on a biologically inspired natural selection process. This algorithm is a subset of evolutionary algorithms used for computings. The GA uses genetic and natural selection principles to solve problems [107]. The parameters used in the GA are shown in the Figure 10. Recent studies have been conducted on the GA algorithm [108], [109], [110], [111], [112]. A concise tabulation of advantages and disadvantages of different search strategies are illustrated in Table 1.

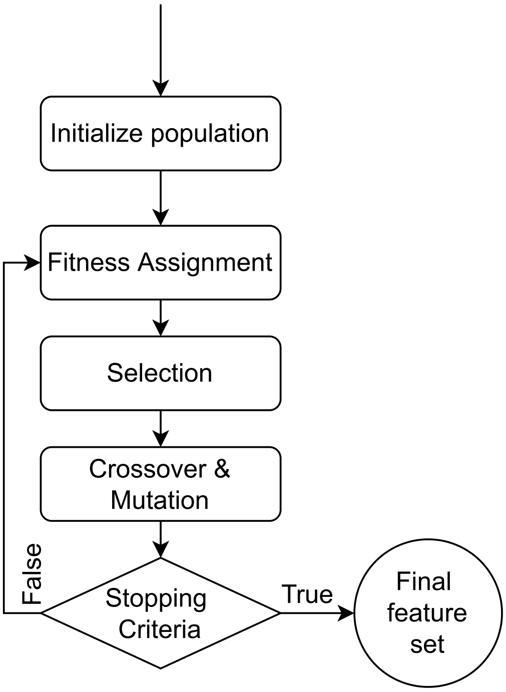


FIGURE 10. The graph illustrates overall working flow of the genetic algorithm.

III. FEATURE SUBSET EVALUATION CRITERIA

An evaluation criterion is a process that aims to find the relevant feature from the feature sets by utilizing various methods. Feature selection has four evaluation criteria: filter, wrapper, embedded, and hybrid. The following section discussed these methods along with their advantages and disadvantages.

A. FILTER METHOD

Filter methods are commonly employed as independent preprocessing methods. Instead, features were selected based on their correlation scores with the outcome variable in various statistical tests. The term "correlation" refers to a purely subjective concept. Furthermore, the classification algorithm does not influence the evaluation of the subsets. To calculate features, several parameters - such as correlation, gain Ratio, Euclidean distance, and others are utilized. These parameters are discussed in the following section and the structure of the filter method is Illustrated in Figure 11.

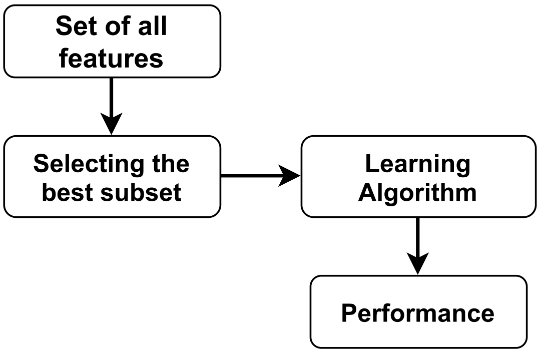


FIGURE 11. The general Structure of filter method.

1) MUTUAL INFORMATION (MI)

Mutual Information (MI) is a statistical technique employed in FS. From equation 1, MI is a metric for determining how two variables (a, b) are interdependent [113]. It assesses the "measure of data" collected on a random variable through the other random variable.

$$I(A, B) = \sum_{b \in B} \sum_{a \in A} p(a, b) \log \left(\frac{p(a, b)}{p(a)p(b)} \right) \tag{1}$$

where $p(a, b)$ is the joint probability function of A and B. $P(a)$, and $P(b)$ are the marginal probability distribution functions of A and B respectively. This equation is used to determine the MI between two discrete random variables, a and b. The summation is performed using a double integral for continuous random variables.

$$I(A, B) = \int_B \int_A p(a, b) \log \left(\frac{p(a, b)}{p(a)p(b)} \right) dadb \tag{2}$$

Statistical measures were used to assign scoring values to each feature in the filter technique. The features were sorted in descending order according to their rankings. A subset of features was selected based on the threshold values. Using the filter approach to select the best features requires less computational time. Because the connection between independent variables is not considered when selecting features, irrelevant features are chosen. Recent studies have utilized MI techniques in their research [114], [115], and [116].

TABLE 1. Advantages and disadvantages of various search strategies.

Category	Algorithms	Advantages	Disadvantages
Exponential Algorithm	Exhaustive Algorithms	Less computational time	Unsuitable for big dataset
	Complete Search	Reduce computational time and consume less memory	Face difficulties to reach optimal goals
Sequential Algorithm	SFS	Less computational time and simple implement process	Presence of redundant features and nesting effect
	SBS	Less computational time and simple implement process	Presence of redundant features and nesting effect
	SFFS	Reduce nesting issues and redundant features	Hard to find all subsets
	SBFS	Reduce nesting issues and redundant features	Hard to find all subsets
	$LminusR$	Overcome nesting issues	Hard to predict values of L and R
	BDS	Reduce requiring time and less memory capacity	Goal state should be known before search
Random Search Algorithm	RGSS	Apply additional algorithms	Relies on random parameters
	Simulated Annealing	Deal with arbitrary systems and assure optimal solution	Expensive cost function and few local minima decline performance
	Random Hill Climbing	Can solve pure optimization problems	Tendency to become stuck at Local maxima or foothills, a plateau or a ridge
	GA	Avoids getting trapped in local optimal solution like traditional methods	Cannot perform well with complexity
	MA	Reduce the premature convergence	It is an extension framework
	LV's	Can determine the correct answer rapidly	Vulnerable to large datasets
	BnB	Less computational time	Cannot perform better in a large dataset

2) PEARSON's CORRELATION (PC)

Pearson's Correlation (PC) is a filter-based method. PC is used to detect the linear relationship between the two continuous variables, X and Y. Its value varies from -1 to $+1$ [117], [118].

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{3}$$

3) CORRELATION COEFFICIENT

The features that show redundancy are dealt with using correlation-based feature selection [119]. The correlation coefficient is used to select features that are highly related to the target variable but have minimal inter-correlation between them [120]. The correlation of each set of features determines the highest correlation coefficient value and immediately selects a feature [121].

4) INFORMATION GAIN (IG)

Information Gain (IG) is filter feature selection method utilized to determine essential qualities from a group of features. When the value of the feature is unknown, IG reduces the risks associated with selecting a class attribute [122]. It is primarily concerned with information theory. It is used to rank and select top features before the learning process begins to reduce the feature size. The entropy value of the distribution was calculated by ranking to estimate the uncertainty of each feature based on its significance in defining separate classes [123]. The entropy of the distribution, sample entropy, and predicted model entropy of the dataset determines the ambiguity [124]. The information gain about X provided by Y is calculated as:

$$IG(X \mid Y) = H(X) - H(X \mid Y) \tag{4}$$

where,

$$H(X) = -\sum_{i=1}^{k} P(x_i) \log_2 P(x_i) \tag{5}$$

is the entropy of variable X and,

$$H(X \mid Y) = -\sum_{i} P\left(y_j\right) \sum_{i} P\left(x_i \mid y_j\right) \log_2 \left(P\left(x_i \mid y_j\right)\right) \tag{6}$$

is the entropy of X after observing another variable Y.

5) GAIN RATIO

The gain ratio is required to improve the IG's bias towards features with high diversity values [125]. The gain ratio is significant when the data were evenly distributed. It is low if all data are directed to only one branch of the property. The gain ratio is an attribute determined by the number and length of the branches. It attempts to correct IG by taking intrinsic information into consideration [126]. The entropy distribution of the instance value can be used to estimate the intrinsic information of a specific feature.

$$\text{Gain Ratio}\,(y, x) = \frac{\text{Information Gain}\,(y, x)}{\text{Intrinsic Value}\,(x)} \tag{7}$$

where,

$$\text{Intrinsic Value}\,(x) = -\sum \frac{|S_i|}{|S|} * \log_2 \frac{|S_i|}{S} \tag{8}$$

Here, $|S|$ is the number of possible values that feature x can take, while $|S_i|$ is the number of actual values of feature x.

6) LAPLACIAN SCORE (LS)

The Laplacian score [127] is a prominent unsupervised feature selection method that estimates features based on location preservation. In other words, a conventional feature is identified if two data points are confined to the present dimension similar to the original space. Consequently, a good feature maintains the local geometrical formation of the data. The Laplacian score (L_r) is expressed as:

$$\mathbf{L}_r = \frac{\tilde{\mathbf{f}}_r^T \boldsymbol{L} \tilde{\mathbf{f}}_r}{\tilde{\mathbf{f}}_r^T \boldsymbol{D} \tilde{\mathbf{f}}_r} \tag{9}$$

where a diagonal matrix is denoted by D, Laplacian matrix defined as $L = D - S$ and f_r is determined as follows:

$$\tilde{\mathbf{f}}_r = \mathbf{f}_r - \frac{\mathbf{f}_r^T \boldsymbol{D}\mathbf{1}}{\mathbf{1}^T \boldsymbol{D}\mathbf{1}} \tag{10}$$

where $1 = [1, \ldots, 1]^T$. The relevant features were sorted in ascending order of L_r after the Laplacian score for each feature was calculated [128].

7) FISHER SCORE

Fisher score is a popular supervised method for selecting features that compute individualized Fisher scores over the data space [129]. Fisher's criterion does not recognize combined effects or handle the similar features but provides optimal predictors [130] under certain orthogonality assumptions. The fundamental premise of the Fisher score is to increase the distances between data samples in different classes while decreasing the distances within the same class. Several recent studies utilized the fisher score filter method for feature selection [131], [132], [133].

8) CHI-SQUARED

The chi-squared (X^2) statistic was used to evaluate the independence of two variables by calculating a score that indicated the independence they are. X^2 measures the independence of the features for the class in feature selection. Before calculating a score, X^2 relies on the assumption that feature and classes are independent [134]. A substantial score value indicates a highly dependent connection.

$$\chi^2(r, c_i) = \frac{N\left[P(r, c_i)\, P(\bar{r}, \bar{c}_i) - P(r, \bar{c}_i)\, P(\bar{r}, c_i)\right]^2}{P(r)P(\bar{r})P(c_i)\, P(\bar{c}_i)} \tag{11}$$

where N signifies the complete dataset, r indicates the presence of a feature (r its absence), and ci refers to the class. Where $P(r, ci)$ is the probability that feature r occurs in class ci. $P(r)$ is the likelihood that a feature resembles the dataset. Some researchers have used the chi-squared filter method for feature selection [135], [136].

9) CORRELATION-BASED FEATURE SELECTION (CFS)

Correlation-based Feature Selection (CFS) is an essential filtering technique that ranks feature subsets using a heuristic evaluation function based on correlation [6]. The evaluation function favors subsets with attributes that are substantially correlated with the class but uncorrelated with one another. This technique avoids the irrelevant features because of its low correlation with the class. The Redundant features should be filtered otherwise, they will be substantially associated with one or more of the remaining features. The validation of a feature is determined by how well it anticipates classes in portions of the instance space where other characteristics have not yet been indicated.

10) FAST CORRELATION-BASED FILTER (FCBF)

The fast correlation-based filter (FCBF) begins with a comprehensive set of characteristics. The fast correlation-based filter computes the feature dependency by employing symmetrical ambiguity and eliminates superfluous features using the backward selection approach [137]. This technique includes an internal criterion that prevents features removal. Different approaches to feature selection are slower than rapid correlation-based filters. The FCBF method algorithm was developed in [124].

11) CONSTRAINT SCORE

The constrain score is a supervised feature selection approach that evaluates features using paired constraints [138]. The features with the highest constraint-preserving ability were selected using this strategy. If it is necessary for two data samples to be linked, they must be close to each other on an excellent feature. If there is a constraint on a good feature between two data samples, the samples must be far apart. A recent study based on the constraint score was conducted [139].

12) RelieF

The fundamental RelieF algorithm [140] calculates the attribute performance by focusing on how well its values distinguish between samples that are close in proximity. RELIEF searches for two nearest neighbors: one from the same class and another from another class. Based on the values, the performance estimate for all features is then updated. RELIEF can deal with both discrete and continuous features, but it is only useful for two-class issues. RelieffF [141] is an enhancement that not only handles multi-class problems but is also more resilient and capable of dealing with missing and noisy data. The RReliefF method [142] was developed when ReliefF was used for continuous class (regression) problems. The Relief family of techniques is particularly appealing because it can be used in a variety of situations. It has low bias, incorporates feature interaction, and can capture local dependencies that other approaches overlook.

13) MINIMAL-REDUNDANCY-MAXIMAL-RELEVANCE (mRMR)

Minimal-redundancy-maximal-relevance (mRMR) is a multivariate filter method that uses a relevant criterion to choose features with the maximum dependency on the target class [143]. A measure is used to eliminate redundancy between the characteristics, which is specified as follows:

$$mRMR\left(F_j\right) = \max_{F_j \in F \setminus S}\left[I\left(F_j; C_k\right) - \frac{1}{m-1}\sum_{F_i \in S} I\left(F_j; F_i\right)\right] \tag{12}$$

where, $I(F_j; C_k)$ is the mutual correlation between feature X_j and class C_k, and $I(F_j; F_i)$ is the correlation between features F_i and F_j. S stands for the selected feature set, and m represents its size (i.e., $m = |S|$). Several studies have utilized the mRMR process [144], [145].

Filter methods instantly select the most consistent features from the data. Features were evaluated based on intrinsic data attributes rather than a clustering algorithm to guide the search for relevant features in the filter method. The filter method is also classified into two ways [146].

1) **Univariate Filter method:** Ranking-based unsupervised feature selection approaches are known as univariate methods. Univariate techniques employ criteria to evaluate each feature individually, resulting in an ordered ranking list of features from which the final feature subset is selected. These approaches successfully identify and remove unnecessary features. However they cannot remove the same features because they do not account for the possible feature dependencies. Alternatively, univariate filter techniques only assess the characteristics separately, ignoring redundancy [147], [148], [149].
2) **Multivariate Filter method:** Multivariate techniques consider feature correlation in their analysis and can therefore manage both irrelevant and duplicated data. Consequently, they identify more than two-way correlations within the feature set these techniques are considered more generic. Multivariate filtering methods evaluate the significance of the characteristics collectively rather than individually. Learning algorithms that use a subset of attributes picked using multivariate techniques are more accurate than others in many instances, but they are computationally wasteful [150], [151], [152].

Table 2 represents different Studies using filter method to select features.

B. WRAPPER METHOD

Wrapper methods evaluate the relative utility of feature sets based on the prediction performance of a learning machine. Classification error rate estimation and theoretical performance constraints are frequently used to evaluate a model's performance. The lower the error rate of feature subset the better the result. An exhaustive search can be conducted when the number of features is small. However, examining all subsets is NP-hard and is subject to overfitting. Sequential forward selection or backward elimination, best-first, branch-and-bound, simulated annealing, and genetic algorithms are just a few of the greedy search strategies that can be implemented [162]. Several of the are very common in the sequential search included in section II. The structure of the wrapper method is shown in Figure 12. The other wrapper methods are discussed in the following section.

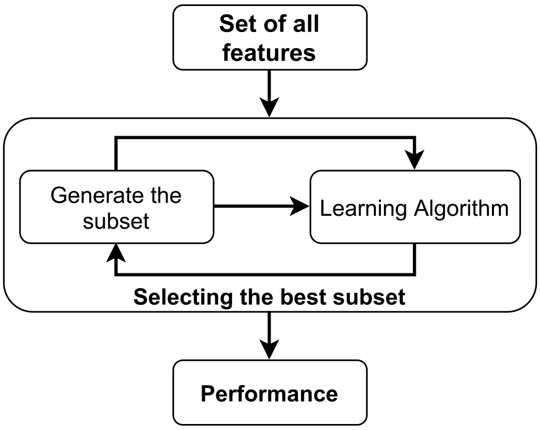


FIGURE 12. The illustration shows the Structure of the wrapper method.

TABLE 2. A table of different works under filter method.

Algorithm	Dataset	Results	Limitation
ReliefF, CBFS, FCBF and INTERACT [153]	Synthetic Datasets	When 40 relevant features are used, CFS success rate was higher than others.	Adequate selection of features is must to improve accuracy and efficiency of classifier methods. The datasets need to be more defined. This system needs to be testified in a benchmark dataset.
K-means clustering [154]	Synthetic, benchmark, and real datasets shows its effectiveness	93433.9 clusters found with filter method.	In clustering, a wrapper method evaluates the candidate feature subsets by a clustering algorithm. The result evaluation can be described in other calculations. A comparison table of other algorithms will be good to understand the difference.
Discretizer + filter [155]	DNA microarray data	95% on binary data, 87.50% on multiclass data.	High number of gene expression contained and the small sample sizes which is a challenging issue. This system needs to be testified in a large dataset.
RFS [156]	12 medical datasets	19.25±6.78 for one class.	High dimensional datasets with a relatively small number of Instances need to be solved. The supervised discretization and selection procedures need to be developed.
22 filter methods [157]	16 high dimensional classification datasets	4% of features.	Multi-criteria tuning needs to be performed with respect to all performance criteria at the drawback of a much more complicated aggregation of the results. This approach could be transferred to model selection.
Entropic Filtering Algorithm (EFA) [158]	5 public domain microarray datasets	Accuracy on three different dataset 0.96, 0.97, 0.90.	The result should be counted in average from all dataset together. Their performance can be extended by utilizing other algorithms.
PLS [159]	DNA microarray	95% Accuracy.	Statistical dependence measures need to be presented for selection in the context of classification. The comparison with other approaches would help to understand the efficiency. This system needs to be testified in a large dataset.
MWMR [160]	PCMAC dataset	83.60% Accuracy.	One major problem in applying DNA microarrays for classification is the dimension of obtained datasets. Multiclass selection problems need to be solved. This system can extend their performance utilizing other algorithms.
Filtered and supported SFS [161]	10 datasets which are from the widely used University of California, Irvine (UCI) repository of machine learning	Accuracy is 99.3% on WDBC dataset.	High-dimensional feature vectors impose a high computational cost as well as the risk of “overfitting”. The computational complexity needs to be reduced and the classifier’s generalization ability needs to be improved.

1) RECURSIVE FEATURE ELIMINATION

Recursive feature elimination (RFE) is a well-known feature selection algorithm. It is popular since it is simple to set up while using, and good at identifying features in a training dataset that are more relevant in determining the desired variable [163]. It is a recursive procedure that sorts the features based on feature importance and an underlying random forest classification model. When using RFE, there are primarily two configuration options: the number of features to choose from and the algorithm used to assist in feature selection. Both of these hyper-parameters can be investigated, but their correct configuration has no significant effect on the performance of the method. This method has been used in the several recent studies [164], [165], [166], [167].

2) BORUTA ALGORITHM

The Boruta algorithm is a wrapper for the random forest classification algorithm in the random forest R package [168]. The random forest classification process is fast, can typically be performed without parameter modification, and provides a numerical estimate of feature importance. It is an ensemble method in which several unbiased weak classifiers, such as, decision trees, vote on classification. These trees were generated one at a time on different bagging samples from the training set. The loss of classification accuracy caused by the random permutation of attribute values between instances is used to calculate an attribute's relevance. It is calculated separately for each tree in the forest and is classified using a specific property. The average accuracy loss' and standard deviation were then calculated. Boruta was built using the same principle as that of the random forest classifier. By introducing randomness to the system and gathering data from an ensemble of randomized samples, the influence of random fluctuations and correlations can be mitigated [169]. Studies have suggested usingg the boruta algorithm for FS [170], [171], [172], [173].

Table 3 presents different studies using the wrapper method to select the features.

C. EMBEDDED METHOD

Feature selection is integral to the learning algorithm for the embedded approach, which continuously develops a classifier and selects a subset of features [182]. These methods frequently function by introducing a sparsity-inducing regularization or prior into the objective function of the learning algorithm, causing the weights assigned to a feature set to be zero. Furthermore, embedded techniques are described as a trade-off between wrappers and filters as well as embedded feature selection in the process of the learning algorithm. As a result, the wrapper and filter methods were used. Furthermore, they like wrappers and, work in conjunction with learning algorithms. Furthermore, they are far more effective than wrappers since they do not need to repeat the learning method. Embedded techniques are frequently unable to provide better learning results than wrappers [183]. The structure of the embedding method is illustrated in Figure 13. The least absolute shrinkage and selection operator (LASSO) and RIDGE regression are two popular implementations of this approach. Both had built-in penalization factors to reduce overfitting. Several studies have employed feature selection [184], [185], [186], [187], [188].

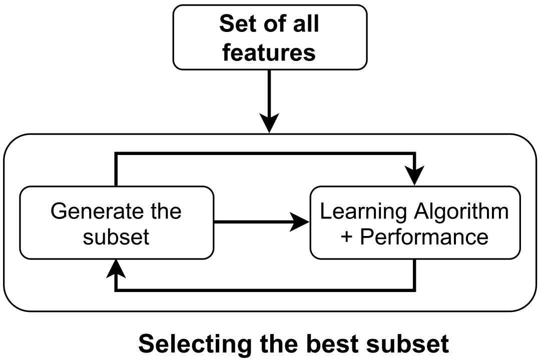


FIGURE 13. Structure of embedded method.

1) LASSO

The least absolute shrinkage and selection operator (LASSO) was developed by Robert Tibshirani in 1996 [189]. This is a beneficial technique because of its two characteristics: regularizing and selecting features. The LASSO technique requires that the sum of the absolute values of the model parameters be less than a particular value (upper bound). This approach penalizes the regression variable coefficients by decreasing some of them to zero via a shrinking procedure known as L1 regularization. Variables with non-zero coefficients after downsizing were selected as part of the model during the FS stage. The goal of this approach is to minimize prediction errors as much as possible [190]. The LASSO system can produce a highly accurate forecast while reducing the variance without considerably increasing the bias by shrinking and deleting coefficients. LASSO is useful, with a limited number of instances and a wide variety of features. Furthermore, LASSO reduces overfitting by removing external variables that are not associated with the response variable, thereby improving model interpretability [191]. Table 4 presents different studies using the embedded method to select the features.

D. HYBRID METHOD

Hybrid feature selection methods have been a subject of great interest in recent decades. The hybrid model attempts to combine the strengths of the two models by utilizing their distinct evaluation criteria in various phases of the search process. Hybrid techniques aim to combine the benefits of wrappers and filters. Two hybridization strategies are [201] commonly used to combine the wrapper and filter methods. Jihong *et al.* [202] proposed a hybrid feature selection (HFS) technique that uses both filter and wrapper models of feature subset selection and focuses on selecting a sub-feature set where all

TABLE 3. A table of different works under wrapper method.

Algorithm	Dataset	Results	Limitation
BDSFS [174]	Mushroom databases	98.01 ± 2.50.	The careful analysis of arguments for both methods was done to identify the best method. Their performance can be extended by utilizing other algorithms on multiclass problems.
Best-first search [175]	Real world dataset	The average accuracy went up from 87.01% to 87.60%, a 4.5% reduction in error	Compound operators would be better to change the topology of the search space dynamically to better utilize the information available from the evaluation of feature subsets. This system needs to be testified in a benchmark dataset.
Support Vector Machines with kernel functions, sequential backward selection [176]	Four real-world data sets	98.25 ± 2.0, 85.70 ± 5.6, 75.54 ± 3.6, 69.33 ± 1.0 on four datasets	The computational complexity needs to be reduced and the classifier's generalization ability needs to improve. High-dimensional feature vectors impose a high computational cost and a high cost of data acquisition. A low-dimensional representation reduces the risk of overfitting.The datasets need to be balanced.
SBS, SFS, SFFS, and SBFS [177]	Comprehensive GIS database, and NDVI	mmce = 0.12 and AUC = 0.92	The various sources of nitrate pollution need to be recognized and the system dynamics need to be understood and fundamental need to be understood. Their performance can be extended by utilizing other algorithms on multiclass problems.
BIRS (best incremental ranked subset) [178]	4 public gene expression datasets of colon, leukemia, lymphoma, and GCM	BIRS chooses the genes M84526, M27891, M31523 and M23197 among the top 20 genes of the ranking and M36652.	The main motive was to figure out the possible combinations in between each procedure search and each attribute measure with less computational complexity and cost, redundant or irrelevant genes, estimation degradation in the classification error. Their datasets need to be described.
Feature subset selection (FSS) [179]	16 datasets.	Accuracy on one dataset 0.96.	Their focus was on the application of wrapper FSS to high-dimensional datasets, in particular datasets with a very large number of variables and a small number of instances. Their performance can be extended by utilizing other algorithms on multiclass problems and on (semi) big data.
SBW FS [180]	A publicly available dataset collected for Chinese FDI.	GSVM with SBW FS performed the best mean accuracy, i.e., 79.55%, among all models with various feature subsets.	The use of genetic algorithms in wrappers is that the output feature subset is not the same when the approach is implemented several times. The datasets needs to be described.
Moth-flame Optimization (MFO) [181]	8 datasets with diverse characteristics	Accuracy is 98.96% on WDBC dataset.	The other methods work on overall characteristics of the data regardless of the classifier selecting the valuable features. High-dimensional feature vectors impose a high computational cost and a high cost of data acquisition. This system can extend their performance utilizing other algorithms.

TABLE 4. A table of different works under embedded method.

Algorithm	Dataset	Results	Limitation
kernel-penalized SVM (KP-SVM) [192]	Four real-world benchmark datasets	76.74 ± 1.9, 97.55 ± 0.9, 96.57 ± 5.6, 99.73 ± 1.0 on four datasets.	Relevant features need to be selected simultaneously during classifier construction by penalizing each feature's use in the dual formulation of SVM. Their performance can be extended by utilizing other kernel functions like polynomial kernel or with weighted SVM.
Infinite feature selection (Inf-FS) [193]	13 benchmarks of cancer classification and prediction on genetic data	91% on one dataset	Filter-based feature selection has become crucial in many classification settings, especially object recognition, recently faced with feature learning strategies that originate thousands of cues. The relations among the feature needs details description.
Greedy Forward Selection (GFS) [194]	4 public gene expression datasets for breast cancer prognosis	AUC average 0.65(0.15)	Biomarker discovery from high-dimensional data is a crucial problem with enormous applications in biology and medicine. Their performance can be extended by utilizing other algorithms on multiclass problems.
MRMR [195]	6 gene expression datasets	1-2% error rate	Need more balanced coverage of the space and capture broader characteristics of phenotypes. The results can be defined in other functions.
MRMR and SVM-RFE [196]	4 public gene expression datasets of colon, leukemia, hepato, and prostate	92.68 ± 5.13, 99.75 ± 1.31, 88.16 ± 5.72, 98.67 ± 3.96 on 4 datasets.	Redundancy among the genes needs to be solved. Figures can be utilized to describe the method more easily. Multi-fault diagnosis in bearings. This system can extend their performance utilizing other algorithms.
ESFS [197]	The Berlin emotional speech database.	Accuracy 72.80%.	The complexity of the classifier parameters adjustment during training increases exponentially with the number of features. Too many input features may lead to the so-called "curse of dimensionality". A hierarchical classifier can be used to separate classes by first separating classes far away from each other and then concentrating on closer classes.,
EUFS [198]	6 publicly available benchmark datasets (ALLAML, COIL20, PIE10P, TOX-171, PIX10P, Prostate-GE).	Clustering results, (ACC%±std) for 6 datasets are 73.6±0.00(100), 63.4±5.47(100), 46.4±2.69(50), 49.5±2.57(100), 76.8±5.88(150), 60.4±0.80(100)	The lack of label information, the vast majority of these algorithms usually generate cluster labels via clustering algorithms and then formulate unsupervised feature selection as sparse learning based supervised feature selection with these generated cluster labels. In NMI results their work needs to be more precise.
Weighted gini index (WGI) [183]	The statlog (ladsat satellite) and letter recognition datasets	AUC of two datasets 0.085 and 0.999.	Imbalanced data is one type of datasets that are frequently found in real-world applications. For this type of datasets, improving the accuracy to identify their minority class is a critically important issue. Determining the optimal weight in GI-FSw remains unsolved.
KP-SVDD and KP-CSSVM [199]	12 highly imbalance microarray datasets	AUC 99.5 on BHAT2 dataset.	The dataset sometimes exhibits significantly, but sometimes they are extremely imbalanced. Data resampling rebalances a dataset artificially by constructing a training set in which all classes can be shattered adequately by standard classification approaches. Cost-sensitive learning and one class learning are also needed to take care of.
Standard 2-norm SVM and linear 1-norm SVM [200]	"Ionosphere" Dataset	Error only rises to 14% from 11%.	Relevant elements should be chosen all the while during classifier development by punishing each component's utilization in the double detailing of SVM. Their performance can be reached out by using other part works like polynomial bit or with weighted SVM.

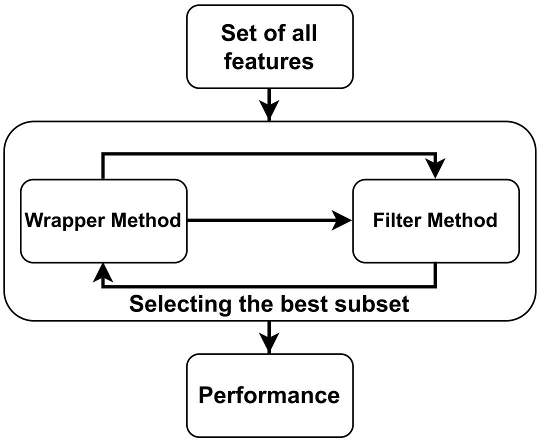


FIGURE 14. Structure of hybrid method.

the selected features are coalitional and significant. Hybrid methods attempt to balance efficiency (computing effort) and effectiveness by combining the benefits of the filter and wrapper approaches (quality in the associated objective task when using the selected features). The structure of the hybrid method is shown in 14. Several recent studies have employed a hybrid method [203], [204], [205].

Table 5 presents different studies using the hybrid method to select the features.

IV. FEATURE SELECTION BASED ON LEARNING METHODS

In machine learning, feature selection is also known as attribute, variable, and feature subset selection. The feature selection strategy tends to be grouped into three machine learning categories based on the availability of class information. The FS learning, supervised, unsupervised, and semi-supervised methods are described below:

A. SUPERVISED LEARNING METHODS

The supervised technique finds a feature subset using labeled data while considering predetermined criteria for determining the relevance of the features. By constrast, unsupervised algorithms seek to discover the inherent data structure to select the most important aspects without assuming prior knowledge [57]. This function locates relevant features based on class labels. This method almost always leads to an overfitting problem owing to the presence of imbalanced datasets. Among the most frequently used supervised feature selection methods are: the Fisher score [60], Hilbert-Schmidt Independence Criteria (HSIC) [61], Fisher Criterion [62], Pearson Correlation Coefficient [63], Trace ratio criterion [64], and mutual information [38]. Several supervised learning methods are explained in section 3, and others are included in the following section.

1) HILBERT-SCHMIDT INDEPENDENCE CRITERION (HSIC)

While reproducing kernel Hilbert spaces (RKHS) [214], [215], an independent criterion called the Hilbert-Schmidt norm of the cross-covariance operator was proposed. Different applications including independent component analysis [216], sorting/ matching [217], supervised dictionary learning [218], and multiview learning [219], have mentioned the proposed measure known as the Hilbert-Schmidt independence criterion (HSIC). According to HSIC, two random variables, x, and y are independent if any bounded continuous function of the two random variables is uncorrelated. HSIC is one of the criteria for detecting non-linear connections that do not require generalized eigenvalue problems or rely on regularization parameters [220], [221].

B. UNSUPERVISED LEARNING METHODS

Unsupervised feature selection (UFS) approaches are extensively used to analyze high-dimensional data. These techniques use unlabeled data owing to the scarcity of promptly available labels. The majority of existing UFS techniques concentrate on the importance of features in preserving the data structure while ignoring feature redundancy [222].

1) UFS WRAPPER METHOD

Wrapper approaches use the results of precise clustering algorithms to evaluate the feature subsets. The discovery of feature subsets distinguishes between these methods based on the aforementioned approach. The quality of the results of the clustering algorithm used for selection was improved in this manner way.

1) Sequential methods: In these methods, the features are sequentially added or removed. [223], [224], [225] are profound works on this topic.
2) Bio-inspired methods: Bio-inspired methods attempt to introduce unpredictability into the search process in order to avoid local optima. Some studies on these methods are presented in [226] and [227].
3) Iterative methods: Iterative approaches resolve the UFS issue and reduce the need for combinatorial search by redefining it as an evaluation problem. [228], [229], [230] are some studies on these methods.

2) UFS HYBRID METHOD

Hybrid-based methods attempt to use the strengths of both, filter and wrapper, to achieve a suitable balance of computational efficiency. It also demonstrates the productivity in the associated objective task when the selected features are used. Hybrid-based methods include a filter frame in which features are ordered or chosen using a measure based on the inherent attributes of the data.

C. SEMI-SUPERVISED LEARNING METHODS

Semi-supervised learning [231] studies a small amount of labeled data and many unlabeled data. Semi-supervised based feature selection methods are distributed into two groups and explored in depth from two different prospectives [232], [233]. FS taxonomy was initially centered on and classified into semi-supervised feature selection procedures based

TABLE 5. A table of different works under hybrid method.

Algorithm	Dataset	Results	Limitation
Maximum Class separability (MCS) feature distribution analysis method [206]	Acoustic Emission (AE) signals	For all Five datasets approx 98.50%.	Multi-fault diagnosis in bearings. This system can extend their performance utilizing other algorithms and can find the other possibilities.
Hybrid feature selection model discriminant feature distribution analysis-based feature evaluation method [204]	Acoustic emission (AE) signals	95% accuracy	Fault diagnosis scheme from bearing data. Relevant elements should be chosen all the while during classifier development by punishing each component's utilization in the algorithm.
Hybrid feature selection [207]	UCI Machine Learning Repository	Average Classification Accuracy: 96.92%.	Human activity recognition. This system needs to be testified in a large dataset.
Hybrid method that combines the filter and wrapper methods [208]	Breast Cancer Wisconsin (Diagnostic) dataset; Breast Cancer Wisconsin (Prognostic) dataset; and SPECTF Heart dataset	AUC of ROC 0.997, 0.774, 0.832 on 3 dataset	High dimensionality in biomedical data classification needs to be reduced. Wrapper methods tend to have superior classification accuracy but require great computational power.
MIM and SVM-RFE [209]	5 different SNP datasets	96% classification accuracy.	Achieving high classification accuracy in such a high dimensional space is crucial for successful diagnosis and treatment.
GADP and X^2-test [210]	6 different micro array datasets.	87.04% accuracy on GCM dataset and 100% accuracy on the rest of 5 datasets.	A proper number of the most relevant genes need to be selected for data analysis. Relevant elements should be chosen all the while during classifier development by punishing each component's utilization in the GADP and X^2-test.
CFS and TGA [211]	DNA microarray data	On Lung Cancer datasets 195.2±5.41 of genes were selected.	To reduce redundant features effectively and achieve superior classification accuracy.This system needs to be testified in a large dataset.
F SSFS [212]	Taiwan Economic Journal database	87.3% accuracy.	The choice of feature variables has a critical impact on the performance of the resulting system. It is needed to investigate to develop a structured method of selecting an optimal value of the parameters in SVM for the best prediction performance. It is also required to to the generalization of SVM on the basis of the appropriate level of the training set size and gives a guideline to measure the generalization performance.
The combination of the filters and the wrappers [213]	DisProt database, Protein Data Bank	99.45% accuracy	Through the filters are very efficient in selecting features, they are unstable when performing on wide feature sets. For instance, microarrays, transaction logs, and web data are all very wide datasets with a huge amount of features.

TABLE 6. Advantages and disadvantages of filter, wrapper, embedded and hybrid method.

Algorithms	Advantages	Disadvantages
Filter	Fast, scalable and independent of classifiers	Ignores dependency and classifier interaction
Wrapper	Interacts with classifier and features	Overfitting problem and high computational cost
Embedded	Avoids overfitting and considers dependency between variables	Consider dependency between features and classifier.
Hybrid	Less overfitting problem and better computational cost	Consider dependency between combination of features and classifier

on their cooperation with the learning process. The second section is based on a taxonomy of semi-supervised FS, which is divided into many categories based on which the semi-supervised learning algorithm is similar to the method used.

1) SEMI-SUPERVISED FILTER METHOD

Semi-supervised filter feature selection approaches analyze the process of learning tasks by examining the intrinsic aspects of labeled and unlabeled data.

- **Based on spectral graph theory and cluster assumption:** Zhao and Liu [234] suggested the spectral graph theory and the cluster assumption for the semi-supervised feature selection method. This approach looks for a cluster with the best consistency with the label information determined by the cluster indicator. This method begins by generating n nodes in a neighborhood graph, similar to the graph created by the Laplacian score. Then, for each feature vector, a cluster indicator was calculated and its significance was assessed by determining two factors. One is whether the indicator's cluster structures are well-formed, and the other is whether the indicator's cluster structures are consistent with the label information.
- **Based on Fisher criterion:** Using Fisher criteria attributes, it selects features with the best discriminant and context abilities. It uses of both labeled and unlabeled data to determine the local structure and distribution. The goal is to improve the ability to distinguish between different classifications using labeled data while maintaining the local structure of the data using unlabeled data. Yang *et al.* [235] proposed a Fisher score-based structure that incorporated a local structure maintaining criterion and a variant strategy. This method uses the local structure and vast distribution information of the labeled and unlabeled data.
- **Based on the Laplacian score:** The principles of the Laplacian criterion and the output of the information for FS are combined into semi-supervised FS methods depending on the Laplacian score [236], [237], [238], [239]. These approaches are graph-based because they produce a neighborhood graph and analyze features to preserve the local structure of the data. The structure is circumscribed based on the learning method of data based on the Laplacian score.
- **Based on pairwise constraints:** It evaluates the significance of features based on their constraint and locality-preserving power using both paired constraints and the local qualities of labeled and unlabeled data [240]. In addition, relevant characteristics must adhere to the data's local structure as well as user-created paired constraints. These methods are classified as graph-based methods since they create two graphs from supervised and unsupervised data.
- **Based on sparse models:** The sparse feature process selects the sparsest and most discriminative features practicing a range of sparse models. The L1-norm (lasso) model is a well-known sparse model. However, the L1-norm model may not always select suitably sparse features. Recent studies [241], [242], [243] have observed that regarding the association between distinct features, grouping features from all data samples together is beneficial. Specific sparse models, such as the l2,1-norm and l2,p-norm, consider feature correlation when selecting essential features from data samples.

2) SEMI-SUPERVISED WRAPPER METHOD

The semi-supervised wrapper FS method was used to forecast the labels of the unlabeled data and examine the effectiveness of the selected feature subset. It uses a single learner or ensemble learning model.

- **Based on a single learner:** A single learner [244] is used to choose a subset of features in a semi-supervised wrapper FS method based on a single learner. It is used to train a classifier that anticipate unlabeled data labels. Subsequently, a subset of the unlabeled data, including the predicted labels, was chosen at random and merged with the labeled data to create a new training set. The learning model and supervised feature selection method select features from the new training set. Following

TABLE 7. Advantages and disadvantages of supervised, Unsupervised and Semi-supervised learning method.

Methods	Advantages	Disadvantages
Supervised	Better performance with labeled data and produce general classification	Prerequisite knowledge is required, Data diversity and the risk of overfitting.
Unsupervised	Gives better performance without any prior knowledge and decrease the gap between input-output	Avoids dependency and avoids correlation between unlikely features
Semi-supervised	Better performance with both labeled and unlabeled data	Only a subset of the training set's output is presented.

random selection, subsets of features were generated and the processes were repeated. The frequency of each feature was evaluated in the feature subsets, and the feature with the highest frequency was merged with the specified feature subset to establish a new feature subset. This procedure was repeated until the size of the feature subset exceeded a predefined threshold.

- **Based on ensemble learning:** Semi-supervised FS approaches choose the anticipated unlabeled data using a confidence metric. The confidence measure is an important factor for determining the success of the semi-supervised FS method in ensemble learning. In such methods, different classifiers are used depending on the training or feature sets. To create different training sets, resampling methods such as bagging are used. By contrast, random subspace methods (RSM) are used to build alternative feature sets. To produce distinct datasets, a blend of resampling and random subspace can be utilized. Several classifiers were trained, and then their output results were combined with ensemble learning methods [245]. Self-training or co-training-based semi-supervised FS methods are based on ensemble learning [246], [247].
    - In the self-training procedure, the fundamental idea is to use labeled data to train a classifier. The classifier is then used to anticipate the labels of data that have not been tagged. Subsequently, a subset of the most confident unlabeled data is chosen and included in the training set, along with its expected labels. This technique is continued when the classifier is retrained on the new training set. Self-training occurs when a classifier uses its predictions to teach itself [248], [249].
    - Co-training is a semi-supervised learning strategy that requires two different feature sets from two different classifiers on the labeled data. Each classifier was provided with features to train with reprocessing to categorize unlabeled data. Other classifiers continuously employ the most confident forecasts of each classifier on unlabeled data as labeled training data [233], [250], [251].

3) SEMI-SUPERVISED EMBEDDED METHOD

Semi-supervised embedded approaches use labeled and unlabeled data to conduct FS during the training process. Semi-supervised embedded feature selection approaches are separated into two categories: those based on sparse models and graph Laplacian and those based on support vector machines.

- **Based on sparse models and graph laplacian:** A range of sparse models and graph-based semi-supervised learning have been utilized to explore labeled, and unlabeled data simultaneously [252]. The most well-known procedure that relies on the graph Laplacian is manifold regularization, which extends several algorithms to semi-supervised approaches [253].
- **Based on Support Vector Machines** Support vector machine-based methods choose features by optimizing the classification margin between classes while utilizing the local data structure. Many strategies, such as manifold regularization, recursive feature removal, merging L1-norm with L2-norm, and replacing L2-norm with L1-norm can be used for SVM-based models [254].

The advantages and disadvantages of supervised, unsupervised and semi-supervised learning method are listed in Table 7 in a concise manner.

Another learning method known as the ensemble learning method, utilizes combination of several learning models. The ensemble learning method is described in the following section.

D. ENSEMBLE LEARNING METHOD

Ensemble learning is a powerful machine learning technique. The basic concept is to improve learning outcomes by combining several learning models [255]. Ensemble learning methods outperform single machine learning models across a variety of machine learning techniques. The rapid growth of ensemble feature selection in recent decades has been based on the concept of ensemble learning. Unlike other feature selection techniques, only one optimal feature subset was selected. The goal of the combination feature selection is to obtain a large number of optimal features. The learning outcomes re set based on several optimal feature subsets and then combined. The most difficult aspect of ensemble learning is deciding which algorithms to use to construct the ensemble and which decision or fusion function is used to combine the results of these algorithms. It is simple to add more algorithms to improve the fusion results. However, the computational cost of adding a new algorithm must be carefully considered. A set of base classifiers must be constructed during the creation process. During the combining phase, the findings of the base classifiers are combined into one. The ensemble concept is at the heart of many well-known machine learning techniques. Bagging, boosting, and stacking are the three most commonly used ensemble models [256]. Some of the most common algorithms are used in these methods. The random forest algorithm is the most commonly used bagging algorithm. There are several algorithms for boosting methods, including AdaBoost [257], the gradient boosting machine (GBM) [258], XGBoost [46], and Light GBM [259]. Govindarajan *et al.* [260] proposed a hybrid RBF-SVM ensemble classification using support vector machine (SVM) and radial basis function (RBF) as primary classifiers. The effectiveness and benefits of the proposed model were demonstrated using NSLKDD datasets. The results indicated that the proposed ensemble RBF-SVM outperformed single-method approaches in terms of effectiveness, with a score of 98.46 percent. Other studies included the ensemble learning method for feature selection [261], [262], [263], [264], [265].

V. RESULT VALIDATION AND PERFORMANCE MEASURES OF FS

Prior knowledge of the underlying dataset's is frequently used to explicitly validate the outcome of an FS process. The relevant and irrelevant feature subsets for a synthetic dataset were identified. The validation result was estimated by determining the relevant and irrelevant features from the feature subset. Such an availability of background knowledge is rare in actual employment. Therefore, researchers must rely on indirect measures, such as observing changes in mining performance as features improve. For example, using the classification error rate as a performance indicator for a learning problem can enable a "before-and-after" investigation. To analyze the error rate of the classifier learned on the entire set of features to the classifier acquired on the selected subset [14], [266].

A. CLASSIFICATION/CLUSTERING ALGORITHM FOR VALIDATION

The evaluation of supervised classifiers such as kNN [267], SVM [268], and Naive Bayes (NB) [269], among many others, utilizes classification accuracy or error rate. Spectral Feature Selection, Statistic-based, and Bio-inspired approaches all use this approach.

1) K-NEAREST Neighbor(KNN)

KNN classifiers using a majority vote of the K-nearest instances, and a new sample is classified. To obtain a regular unweighted KNN algorithm, the parameter kernel must be changed to rectangular. Several studies have utilized KNN classifiers for model validation [270], [271], [272].

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{13}$$

2) SUPPORT VECTOR MACHINE (SVM)

As a decision boundary, Support Vector Machines use the hyperplane in the optimal feature space in terms of the maximum margin concept. Kernel functions change the shape of the hyperplane from linear to non-linear [273]. Support vector machines are frequently used with the RBF kernel. The two hyperparameters are the regularization parameter C and the kernel width parameter. SVM classifiers have been used in recent studies [274], [275], [276]. Other classifiers have been used in recent studies, such as the random forest classifier, Naive Bayes classifier, and c4.5 classifiers. Recent studies that are utilized these classifiers [277], [278], [279], [280], [281], [282].

3) NAIVE BAYES CLASSIFIER

The naive Bayes classifier is a simple and efficient classification method that facilitates the development of a fast ML algorithm's ability to make rapid predictions. It is a probabilistic classifier that generates forecasts based on an entity's probability. The existence of one feature in a class is assumed to be independent of the presence of any other feature using a naive Bayes classifier. The probabilities for each element in the naive Bayes algorithm are determined separately from the training dataset. A search technique is used to assess the efficacy of combining the probabilities of several attributes and forecasting the output variable. There is no built-in method for determining the relevance of features in Naive Bayes classifiers. Naive Bayes algorithms determine the conditional and unconditional probabilities associated with the features, that forecast the class with the highest probability. This can be used to solve multi-class prediction problems. If the assumption of feature independence is maintained, it can outperform the other models while using significantly less training data. For categorical input variables, Naive Bayes was better than number.

4) RANDOM FOREST CLASSIFIER

A random forest comprises a massive set of discrete decision trees that work together as an ensemble. The numerous trees in the random forest individually spit out class prediction. The class with the highest choice was the prediction of the model. It employs bagging and feature randomization to create an interconnected forest of trees, the aggregate prediction of which is more accurate than that for a single tree. The underlying premise of random forest is that many highly interconnected models (trees) acting as a committee will outperform any of the measurements of individual models. Clustering algorithms such as k-means [283], EM [284], and

COBWEB are used to evaluate the findings [285]. Measures like Normalized Mutual Information (NMI) and Clustering Accuracy (ACC) are often used to assess clustering quality.

5) K-MEANS CLUSTERING

K-means clustering is a type of unsupervised learning (data without defined categories or groups). The purpose of the algorithm is to locate groups in the data, where K represent the quantity. The goal is to reduce the within-cluster sum of squares (WCSS) while increasing the between-cluster sum of squares (BCSS) [286]. The most recent work of k-means clustering for feature selection [287], [288], [289].

6) EXPECTATION MAXIMIZATION CLUSTERING

The K-means technique is comparable to the EM (expectation maximization) technique. Rather than allocating samples to clusters to optimize the disparities in means for continuous variables, the EM clustering technique computes the probabilities of cluster membership based on one or more probability distributions [290]. If information is unavailable, the EM technique is used to generate the maximum likelihood parameter estimates. Furthermore, the EM technique can also be used when there are latent data, which is data that was never intended to be discovered in the first place and is hence unseen. Recent studies on the task of feature selection have been conducted [291], [292], [293]. Additional clustering algorithms were employed for the validation of FS. The most recent studies are [294], [295], [296], and [297].

B. VALIDATION MEASURES

Several evaluation measures are often used to evaluate the performance characteristics of feature selection methods. Specificity and sensitivity are frequently used in medical classification, precision and recall in data classification in computer science, as well as the area under the curve in radar signals. Various metrics are used to assess the overall performance of the algorithms. The most frequently used evaluation measures were examined and provided in-depth [298].

- **True Positive (TP) and True Negative (TN):** True positive (TP) is an outcome in which the model forecasts the positive class correctly. The actual results come from the positive classes, and are expected to be positive. A true negative (TN) is an occurrence in which the model correctly predicts a negative class. The actual findings come from the negative class, which is predicted by the model be negative.
- **False Positive (FP) and False Negative (FN):** False positive is a binary classification error in which a test result incorrectly shows evidence of a circumstance such as a disease when it is not present. In contrast, a false negative is the reversed error in which a test result incorrectly demonstrates the absence of a condition when it is present [299], [300].
- **True positive rate (TPR)/Recall/Sensitivity:** TPR is the percentage of all positive samples that are correctly classified [301], [302], [303]. This was calculated using the following equation:

$$TPR = \frac{TP}{TP + FN} \tag{14}$$

Here, TP represents the number of correctly categorized positive instances. In contrast, FN in the TP formula represents the number of positive cases incorrectly classified as negative cases. The percentage of successful cases was equal to the sum of TP and FN.

- **True Negative Rate (TNR)/ Specificity:** TPR is the proportion of actual negative choices to complete the negative observations. TNR calculated as:

$$TNR = \frac{TN}{TN + FP} \tag{15}$$

In the TN formula, TN stands for the number of correctly identified negative instances. By contrast, FP represents for the number of incorrectly categorized negative cases. The number of negative instances is equal to the sum of the TN and FP.

- **Accuracy:** Accuracy is a widely applied metric for assessing classifier performance in text applications. This denotes the proportion of documents in a document set that have been correctly classified. This indicates the categorization model's quality; the higher the number, the better and specifies the percentage of samples that are correctly classifieds. The formula for accuracy is as follows:

$$\text{Accuracy} = \frac{TN + TP}{TP + FP + TN + FN} \tag{16}$$

- **Precision:** The proportion of relevant results is referred to as precision. Actual positive observations divided by the total significantly positive observations are indeed the ratio [301], [304]. The precision is expressed as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{17}$$

- **F-score:** This is a singular score derived from a combination of recall and precision measurements [302], [304]. The F-score is a harmonic mean of the recall and precision metrics that is expressed as:

$$F\ \text{Score} = 2\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

- **Clustering Accuracy:** The clustering accuracy can be calculated by comparing the label derived via clustering with the true label [198].

$$\text{Acc} = \frac{\sum_{i=1}^{n} \delta\left(\text{map}\left(l_i\right), y_i\right)}{n} \tag{19}$$

where, li and yi are $xi's$ cluster and true class labels, respectively, and n is the total number of data points. (x, y) is the delta function that matches 1 if $x = y$ and equals 0 otherwise, and $map(li)$ is the permutation mapping function that outlines each cluster label ri to a similar label from the data set.

- **Error rate estimation:** The ratio of the number of inaccurately predicted output to the total number of data can be termed as the error rate [195], [200]. If the target value is classified, the error is expressed as the error rate. If the summation of two inaccurate predictions $(FN + FP)$ is divided by the summation of a dataset $(P+N)$, the result is actually the error rate of that dataset. The following formula for the error rate is as follows:

$$\text{Error rate } = \frac{FN+FP}{P+N} \tag{20}$$

- **Mean misclassification error (MMCE):** MMCE is calculated using (1-Accuracy). The misclassification rate ranges from zero to one [177]. The formula for MMCE is as follows:

$$\text{MMCE } = \frac{FN+FP}{TP+FP+TN+FN} \tag{21}$$

- **Mean absolute error (MAE):** MAE estimates the difference between the predictions and differ the true probability. It is estimated as:

$$\text{MAE} = \frac{\sum_{j=1}^{M}\sum_{i-1}^{N}|f(i,j)-P(i,j)|}{M\times N} \tag{22}$$

- **Area under the curve (AUC):** The AUC is a traditional measure used to estimate classification performance, determined as the area under the ROC curve [183], [194]. The AUC is a benefit standard. It calculates all possible points underneath a curved line [208]. It divides the curved line into several parts and calculates the AUC by adding the areas of these parts [199].
- **Leave One-Out Cross Validation (LOOCV):** The LOOCV is a procedure to evaluate the effectiveness of those algorithms which are predicting depending on data which are not used to train any model [211]. It works similarly to cross validation.
- **Normalized Mutual Information (NMI):** NMI is a measure that can be used to assess the quality of clusters [176], [178]. The NMI can now be obtained from the following equation given the clustering result.

$$\text{NMI} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{c} n_{ij}\log\frac{n_{i}j}{n_i\hat{n}_j}}{\sqrt{\left(\sum_{i=1}^{c} n_i\log\frac{n_i}{n}\right)\left(\sum_{j=1}^{c}\hat{n}_j\log\frac{\hat{n}_j}{n}\right)}} \tag{23}$$

- **Stability:** Stability measures the robustness of a feature-selection approach [305], [306]. When the training set evolves, robustness implies that the selected characteristics remain stable. This was calculated by using the following equation:

$$\text{Stability } (S) = \frac{2}{m(m-1)}\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} I_J\left(S_i, S_j\right) \tag{24}$$

- **Runtime, training time, and test time:** Runtime is the amount of time it takes to select features. The time required to generate feature weights, rank features using feature weight, and select the top-N features is included in the runtime. Training time refers to the amount of time required to train the classifier. The amount of time required to test a trained classifier may vary owing to differences in the operating, training, and test times of classifiers using different FS methods. We chose these three times to demonstrate efficiency from various perspectives, and these times were cost-dependent.

VI. FEATURE SELECTION ANALYSIS FOR BIG DATA

Big data are defined as ‘‘a dataset whose size exceeds the capability of typical dataset management systems in gathering, storing, processing, and analyzing.’’ It usually has three characteristics: Huge volume, wide variety, and rapid change [1–3]. The challenge posed by these 3V characteristics, namely volume, type, and velocity, have become the focus of learning methods when dealing with extensive data. Furthermore, duplication and relatedness, which are essential in massive datasets to avoid losing valuable content, frequently make the mining procedure more critical. Feature selection (FS) has improved data mining owing to its superior performance in locating correlated features and removing redundant or uncorrelated features from the original dataset [13], [14]. Considering the 3V characteristics, classic feature selection approaches confront three distinct issues in the context of big data:

- Traditional feature selection methods typically require a significant amount of learning time, making it difficult for the processing speed to keep up with the changing of large data;
- In a broad sense, big data contains not only a massive amount of irrelevant and/or redundant features but also possible noises of varying degrees and types, significantly affecting the performance of feature selection;
- Some data are untrustworthy/forged as a result of varied acquisition methods or even losses, considering the complexity of feature selection.

1) SCALABLE GLOBAL MUTUAL INFORMATION-BASED FEATURE SELECTION (SGMI)

SGMI is a distributed and scalable global MI-based feature selection framework that develops a similarity matrix in a single pass and a scalable manner. Subsequently, based on the similarity matrix, it employs a feature ranking algorithm to discover a globally optimal solution. The similarity matrix indicates the dependency among features simultaneously, and it can be computed using the MI or CMI, with the former having less complexity than the latter. The SGMI framework employs three optimization approaches. The first employs a MI similarity matrix, whereas the others use a CMI similarity matrix. In this study, three techniques are developed: SGMIQP, SGMI-SR, and SGMI-TP. Consequently, these methods establish a feature ranking to place informative characteristics at the top of the ranking.

2) DISTRIBUTED CORRELATION-BASED FEATURE SELECTION (DiCFS)

Palma-Mendoza *et al.* Introduced DiCFS-VP and DiCF-Shp, two parallel and distributed variants of the CFS filter-based FS algorithm utilizing the Apache Spark programming model. The first method distributes the data by splitting 545 rows, whereas the second distributes the information by splitting columns, as suggested by Ramrez-Gallego *et al.* [32]. Both DiCFS-vp and DiCFS-hp can handle larger datasets in significantly less time than the traditional WEKA implementation. Furthermore, expensive WEKA memory needs were sometimes the only viable solution for processing specific datasets. Overall, the horizontal partitioning schemes version (DiCFS-hp) proved to be the preferable option because of its better scalability and natural partitioning mode, allowing the Spark framework to use cluster resources better. For classification problems, the benefits of distribution over the non-distribution version are comparable to, if not superior, those already proven for the regression domain [10].

3) DISTRIBUTED QUADRATIC PROGRAMMING-BASED FEATURE SELECTION (DQPFS)

DQPFS, a feature ranking algorithm based on the Apache Spark computing paradigm, is described as a distributed and scalable redesign of the traditional QPFS technique that can cope with Big Data with a considerable number of instances and attributes simultaneously. The computational bottlenecks in QPFS are the redundancy matrix and relevancy vector. The suggested method is not affected by this issue, and it may generate a matrix and vector using independent tasks and indifferent worker nodes. It has a little better scale-out and a slight worse speed-up than DiRelief; however, its execution time is substantially shorter. DQPFS is scalable, and it can analyze large datasets in a short period. In addition to speed-up, scale-out, and execution time, the accuracy of the final outputs of DQPFS and DiRelief were compared using the Naive Bayes classifier. The findings did not reveal that the accuracy of the classifier was generally superior to that of DiRelief. However, they showed that DQPFS could be a suitable choice for feature selection in an extensive dataset.

VII. APPLICATION OF FS

Feature selection is trendy in various fields such as intrusion detection, bioinformatics, medicine, and industry. The application of the FS domain are categorized into several subsections: general, medical, representative, intrusion, and industrial applications. The following section of the study is explained the available applications of the FS domain.

A. GENERAL APPLICATIONS

The feature selection approach has many application domains. Some areas are interrelated with others and some areas have sub-areas. General applications are those where fields are not identified as a whole but are very often used. These domains are categorized in the following subsections.

1) TEXT MINING

The bag-of-words model is a typical method for encoding a document in text mining [307]. The purpose is to model each text based on the number of words that appearing there in it. Typically, feature vectors are built to indicate the count of a single word; however, another option is to confirm the presence or absence of a word without providing a count. A lexicon is a collection of words whose occurrences have been tracked. When a dataset requires expression, words from the documents can be combined to form a vocabulary, which is then reduced by feature selection. During feature selection, it is possible to perform some preprocessing, such as removing rare words with very few instances, removing excessively familiar terms (e.g. ''a,'' ''the,'' ''and,'' and similar), and combining the various inflected forms of an expression (lemmatization, stemming) [308].

1) **Text classification:** Text classification is the process of categorizing text into a set of specified categories or labels. This issue is crucial for spam detection devices connected to the internet, retail and bidding websites. Each word in the document is referred to as a feature. However, this requires far more input features than instances. A portion of the vocabulary must be chosen to, allow the learning process to use less computing, storage, and/or bandwidth. A preprocessing stage is commonly used in feature selection to eliminate unusual terms and integrate them into the same term. There are diverse ways to express feature values, such as using a Boolean value to indicate whether a word counts the number of times it resembles it. The range of possible text documents may still be extensive after this preprocessing stage; therefore, feature selection is critical. In recent decades, several processes have been proposed and applied for this purpose [308], [309], [310].

2) IMAGE PROCESSING

The number of possible image attributes is almost endless; therefore, expressing images is difficult [16]. The chosen features are typically determined by the program working on them. Histograms of oriented gradients, edge orientation histograms, Haar wavelets, raw pixels, gradient values, edges, color channels, etc. are samples of features [311].

1) **Image Classification:** Image classification has become a prominent subject due to effective ways to categorize images into categories. Image features are frequently numerically examined to determine what type of components they are. However, image processing typically requires a significant computational and processing power. Feature selection can reduce the number of characteristics required to accurately identify an image. Although a data explosion has

demonstrated the ability of feature selection algorithms to handle millions of images, the need to know which features to extract from each pixel has existed for decades. Some methods extract textural information from a given image, including Markov random fields and co-occurrence features, which is a prominent issue in this field. For image categorization, several researchers have used the FS method [312], [313], [314]. Automatic image annotation can also benefit from feature selection. Two weighted feature selection techniques [315], [316] have been presented to assist clustering algorithms in dealing with several data dimensions and scaling to highly targeted keywords. Researchers have also attempted to develop automatic feature extraction using image classifiers in high dimensional feature spaces [317], [318].

2) **Image Retrieval:** Feature selection is applied to content-based image retrieval to facilitate quick browsing, searching, and recovery [319]. Content-based image retrieval indexes images based on visual contents by utilizing text-based keyword indexing. The large number of images stored in the database poses the most significant challenge for content-based picture retrieval.
3) **Face recognition:** A complex image recognition task involves recognizing a human face. With its multiple legal and commercial possibilities, face recognition has become one of the most emerging research topics in recent decades. Analyzing selected facial features from an image with features in a facial dataset can determine or authenticate the source. Determining which visual elements are most useful for identification or verification is a critical issue in this field. However, this is a difficult process because object photos have many duplications; additionally, facial datasets have many attributes but few samples. Recently, face recognition FS algorithms have been proposed as solutions to these problems. The FS filter approach is popular because it is computationally more expensive than the wrapper or embedding methods. Some studies [320], [321], [322] employed the FS method for face recognition. Lee *et al.* [323] published a new colored face recognition approach that uses a sequential floating forward search (SFFS) to find the best color features for recognition. Also, it's important to note that various proposed solutions based on evolutionary computation techniques are effective [324], [325], [326], [327], [328], [329].

3) SOFTWARE DEFECT PREDICTION

There are various software quality assurance attributes such as reliability, functionality, fault proneness, reusability, and comprehensibility [330]. Selecting the most appropriate software metrics that are likely to indicate fault proneness is critical.

4) MASS SPECTRA ANALYSIS

Mass Spectrometry (MS) has established itself as a new and appealing framework for diagnosing diseases and protein-based biomarker analysis [331]. A mass spectrum has thousands of possible mass/charge (m/z) ratios on the x-axis, each with its signal intensity value on the y-axis. A typical MALDI-TOF low-resolution proteomic profile can contain up to 15,500 data points in the 500-20000 m/z range. With higher resolution equipment, the number of points can be increased even further. For data mining and bioinformatics purposes, each m/z ratio can be regarded as a separate variable whose value is the intensity.

5) STOCK MARKET ANALYSIS

A variety of stock index futures are available. Financial data, especially stock market data, are too extensive to be searched for [332]. The presence of significant volumes of continuous data, in particular, may make explicit idea extraction from raw data difficult because of the vast quantity of data space governed by continuous features [333]. Consequently, when searching, it is necessary to reduce the dimensionality of the data and eliminate irrelevant components.

6) SENTIMENT ANALYSIS

Natural language processing is used in sentiment analysis to capture variability. It is not merely a categorization based on topics or the computational treatment of individuality, sentiment, and judgment in the text. It can be used in recommendation systems to provide answers to questions [334].The positivity or negativity of an opinion is determined based on many characteristics such as term presence, feature frequency, feature presence, term location, POS tags, syntax, topic, and negation. Not all features are required in every case. Therefore, feature selection is necessary.

7) GENRE CLASSIFICATION

Filenames, authors, sizes, dates, track lengths, and genres are frequently used to categorize and recall materials. Categorization is impossible based on these data; hence the feature selection process is intended. Feature selection in genre classification, refers to the process of converting an audio segment into compact numeric values [335]. Owing to the increased dimensionality of the feature sets, feature selection was used as a preprocessing step before classification to reduce data dimensionality.

8) SEQUENCE ANALYSIS

Bioinformatics has a long history of sequence analysis. Two types of concerns can be recognized in the domain of feature selection: content and signal analysis. The concerns are explained in the following.

1) **Content analysis:** Content analysis explores a sequence's general properties, such as its affinity for coding potential prediction and the capacity to perform a particular biological function. Forecasting of

subsets that code for proteins has been a long-standing problem in bioinformatics. Many different types of Markov models have been developed, including the Interpolated Markov model (IMM) [336], extended IMM framework [337], and Markov blanket multivariate filter method (MBF) [151].

2) **Signal analysis:** The discovery of significant motifs in a sequence, such as gene structural components or regulatory regions, is the objective of signal analysis. Many sequence analysis methods rely on the detection of small, almost conserved signals in the sequence, primarily binding sites for different proteins or protein complexes. A popular method for identifying regulatory motifs is to use regression methodology to link patterns to gene expression levels. The motifs that maximize the fit of the regression model can subsequently be found utilizing feature selection [282], [338]. Ben-Dor [339] inspired another classification approach and used a threshold number of misclassifications (TNoM) to score the genes relevant to tissue classification.

B. REPRESENTATIVE APPLICATIONS

Feature selection is a critical knowledge discovery strategy for data analysis. It has been used in a various fields. Following a discussion of some significant advances in feature selection, we look at some representative applications of feature selection, such as bioinformatics, social media, and multimedia.

1) BIOINFORMATICS

Sequence analysis, microarray analysis, mass spectra analysis, single-nucleotide polymorphism analysis, and text and medical literature mining have all used feature selection. The high-dimensionality of data in bioinformatics [340], for example, has resulted in a plethora of feature selection strategies that have been presented in the discipline. In bioinformatics, feature selection is commonly used to solve the problem of high dimensional small sample size (HDSSS) data. An ensemble feature selection technique wass used to identify biomarkers for cancer diagnosis. This study examines ensemble feature selection strategies employing linear SVMs and Recursive Feature Elimination (RFE) feature selection mechanism. In the first phase, distinct bootstrap sub-samples of the training data are drawn. Then the RFE is implemented in all of these bootstrap sub-samples, yielding a diverse collection of feature evaluations.

2) SOCIAL MEDIA

In recent decades, social media sites such as Facebook and Twitter have grown in popularity. These media also provide a convenient means for people to communicate. The enormous dimensionality of actual social media data creates new challenges for data mining tasks. Feature selection is a way method used to reduce the dimensionality of social media data. Domain knowledge must be incorporated to qualify for feature selection on social media. One of the domain pieces of knowledge considered in the social media world is the link information between users or posts such as tweets, blogs, or photos [341], [342]. However, using this knowledge, two fundamental difficulties in feature selection must be investigated: (1) relation extraction from linked data, including labeled and unlabeled data, and (2) mathematical representation for such relations [343].

3) MULTIMODAL RETRIEVAL

The quantity of multimedia data available on real-world multimedia streaming websites, such as Flickr and YouTube is rapidly increasing. We all are aware that multimedia, such as photographs and movies, can provide us with a variety of advantages. By contrast, the resulting characteristics are frequently over-complete when describing specific semantics. It is critical to improve the interpretability of multimedia data by selecting from a limited set of features [344].

C. MEDICAL APPLICATIONS

People may generate and store data at an unprecedented rate in the modern age. This surge in the amount of data accessible for further analysis is evident in medicine and other fields. Artificial intelligence technologies have been used to solve a variety of medical problems and, automate time-consuming and frequently subjective manual operations performed by practitioners in various specialties.

1) MEDICAL IMAGING

Medical imaging has revolutionized health care, with benefits such as better patient care and earlier diagnosis. Image analysis approaches have been shown to be effective in a various of real-world circumstances. However, because medical datasets typically have many features but only a few samples of a specific condition, feature selection preprocessing is almost always required. Medical images were retrieved using technologies such as X-rays, computed tomography (CT) scans, magnetic resonance imaging (MRI), retinographies, and ultrasound images [345] are analyzed further with image classification or segmentation methods. Many approaches to screening [346], diagnosis [347], and treatment preparation [348] have been presented. These techniques are frequently used in the extraction of features, or for the estimation of image attributes such as, color, texture, or shape. However, some aspects may be redundant for a specific medical conditions when general-purpose approaches are employed. This method, combined with the high dimensionality of the material, requires the use of feature selection algorithms.

2) BIOMEDICAL SIGNAL PROCESSING

Clinical medicine analyzes and measures biological signals for their prevention, diagnosis, and monitoring. However, feature selection methods have many applications in this field because of the large volume of data and the relevance of interpretation. Biomedical signal processing automates the monitoring and analysis of biological signals. Biomedical signals have been automatically generated for diagnosis [349], tracking [350], and rehabilitation [351] purposes. Throughout this sector, researchers have concentrated on developing new signal processing techniques that provide practitioners with real-time data for medical decisions. Depending on the application, these methods entail encoding biological signals using Fourier and wavelet basis functions and auto-regressive parameters. This representation can be considered as a feature vector that can further determine the most relevant attributes and lower the dimensionality of the final dataset.

3) DNA MICROARRAY DATA

In recent decades, biomedicine has been a frequent topic in machine learning because of the large amount of data analyzed from genetic tissues. The proliferation of DNA microarray datasets has aided the emergence of a vibrant field of bioinformatics and machine learning research. Microarray data, with a small number of samples but many features, are typically treated as structured data for machine learning applications. Researchers have been working with microarray datasets using feature selection methods to minimize dimensionality from the beginning. Filters are the most frequently used FS methods because of their independence in the learning method. They are less computationally expensive than the other methods. This is particularly critical when dealing with microarray data. The minimum number of samples can lead to data overfitting, making wrappers unnecessary. The current methods include the minimum redundancy maximum relevance (mRMR) algorithm [352], and temporal minimum redundancy maximum relevance (TMRMR) [353].

D. INDUSTRIAL APPLICATIONS

In industrial applications, where multiple redundant sensors monitor the operation of a tool, feature selection is critical for defect identification. Liu *et al.* [354] demonstrated a method to improve the accuracy of identifying a failure (i.e., solving a binary classification problem of the machine state as faulty vs. regular). They envisioned using a global geometric model and a similarity metric to select features in fault diagnostics. The goal is to identify feature subsets that are geometrically related to the original feature set. These three alternative similarity measures were tested and compared, angular similarity, mutual information, and structural similarity index against FS methods based on distance and entropy and SVM and neural network wrappers.

E. INTRUSION DETECTION

Signature-based, anomaly-based, specification-based, and hybrid intrusion detection methods are divided into four categories depending on the detection mechanism utilized in the system. Signature-based intrusion detection systems are effective and productive for detecting existing threats, and their operation are simple. Signature-Based IDS include the Artificial Immune System (AIS) [355], the Collaborative Block Chained Signature-Based IDS (CBSigIDS) [356], and the IPFIX-based IDS (FIXIDS) [357]. Anomaly-based detection aims to anticipate the system's ''ordinary'' pattern and generate an anomaly alert whenever the discrepancy between an immediate occurrence and the regular pattern reaches a predefined threshold. Hybridized Feature Selection Approach (HFSA) [358], Hybrid Anomaly Detection Model (HADM) [359], and Unsupervised Heterogeneous Anomaly Based IDS [360] are several anomaly-based IDSs. A professional manually build the required pattern, which consists of a sequence of guidelines that compare different valid behaviors of a device, for the specification-based detection approach. If the specifications are sufficiently precise, the pattern may be able to detect illegal patterns of activity. The Finite State Machine (FSM) methodology appears to be appropriate for modeling network protocols [361]. Hybrid detection exploited the strengths of each intrusion detection method while minimizing its flaws and constructing a solid schema to detect the intrusion. A key feature of hybrid detection is the use of a key signature-based detection system in conjunction with an additional anomaly-based model. Signature-Based Anomaly Detection Scheme (SADS) [362], Artificial Bee Colony and Artificial Fish Swarm (ABC-AFS) [363].

VIII. CHALLENGES OF FS

As stated at the commencement of this article, continuous advancements in computer-based technology have revolutionized researchers and engineers to gather information at an ever-increasing rate. To deal with the complexities of studying big data, feature selection is a necessary preprocessing step that must be altered and improved to accommodate high-dimensional data. We discussed the significance of feature selection and recent developments in a variety of application domains. However, dozens of new issues have emerged in the emerging big data environment, indicating current research areas in feature selection.

A. SCALABILITY

Most of the existing learning algorithms are created when dataset sizes were significantly smaller. However, today's small and large-scale learning challenges require distinct solutions. The typical approximation-estimation trade-off applies to small-scale learning. Furthermore, in the case of large-scale learning issues, this trade-off is more complicated not only for accuracy but also for the computing complexity of the learning algorithm. The most serious issue is that, as a result of the big data trend [364], data is becoming increasingly large. This issue can arise in any method, including both supervised and unsupervised feature selection. Currently, the number of characteristics in many fields, such as gene analysis, can easily exceed thousands, if not millions. This raises the cost of calculation and necessitates advanced search algorithms, but these features have their issues. Thus the problem cannot be handled solely by increasing computing capacity. Therefore, it is necessary to develop new approaches and algorithms for this purpose. Scaling up is not only about the dataset. There are additional circumstances in which a researcher can determine the magnitude of a machine learning endeavor intimidating [365], such as

- **Model and algorithm complexity:** Many high-accuracy learning algorithms use either sophisticated, non-linear models or computationally intensive subroutines.
- **Time restrictions for inference:** Sensing-based applications, such as robot navigation or speech recognition, require real-time forecasts.
- **Prediction cascades:** The joint output space for applications that demand consecutive, interdependent predictions is exceptionally complex.
- **Parameter sweeps and model selection:** Various instructional executions are required to adjust the learning algorithm hyperparameters and evaluate their statistical significance.

Scaling up learning algorithms is an essential topic for all these objectives. The influence of increasing the training sample size on the computing results of an algorithm in terms of accuracy, training time, and assigned memory is known as scalability. As a result, the goal is to establish a compromise between these objectives, in other words, to obtain ‘‘good enough’’ answers as ‘‘quickly’’ and ‘‘effectively’’ as necessary [366]. Large-scale feature selection issues [205], [367], [368], [369], [370], [371], [372], where the dimensionality approaches have been suggested by researchers. One of the most common methods for dealing with the scalability issue is to distribute the data across multiple processors. Tan *et al.* [373] presented a new adaptable feature-scaling method that has been applied to a large number of synthetic and natural datasets and allows scalability to massive data scenarios. It is based on feature selection in groups and multiple kernel learning.

B. STABILITY

This is the sensitivity of the selection to data disturbance [374]. In the realm of bioinformatics, experts want to obtain the same or comparable set of genes selected each time they acquire new samples with a small amount of disturbance. Domain experts, however, would be hesitant to recognize these algorithms if they were given drastically different feature sets with minor data disruption. The underlying properties of the data have also been discovered to have an impact on the stability of FS algorithms, suggesting that the stability problem may be data-dependent. These criteria also include the dimensionality of features and, the number of data instances. However, according to Li *et al.* [375], studying the stability of unsupervised FS is far more difficult than studying stability for supervised methods. Unsupervised methods do not have adequate advanced knowledge regarding the cluster structure of data [376]. A few recent attempts have been made to analyze the stability of feature selection approaches in unsupervised scenarios. Much work remains to be done in this area [374].

C. COMPUTATIONAL COST VS PERFORMANCE

Most feature selection approaches are computationally inefficient, which is a critical issue in feature selection because they frequently involve many assessments. Although research has shown that filter approaches are typically more efficient than wrapper approaches, this has not always been the case [377]. Therefore, proposing efficient and effective solutions to feature selection challenges remains a challenge. To reduce computing costs, two primary factors must be considered: an effective search technique and quick assessment measure [78]. As the assessment procedure consumes most of the computing cost in the existing approaches, a quick evaluation criterion may have a higher impact than the search technique. It should be emphasized that the parallelizable nature of Evolutionary Computation makes it suitable for grid computing, graphics processing units, and cloud computing. This nature makes it ideal for grid computing, graphics processing units (GPU), and cloud computing, all of which can be employed to speed things up.

D. DISTRIBUTED FEATURE SELECTION

A feature selection method was employed to address an issue in the past, and a single learning model was used. However, for large-scale data, a single learning model is not recommended because the dataset can be split across multiple processors, each running the same feature selection technique and combining the results. A never-ending stream of large amounts of data can in real-time. If the data are all streamed into a single processor, different parts of the data can be handled by other processors working in parallel. If data are streamed into many processors, it also can be handled in the same way. Although the dataset is not very large, several feature selection approaches must be used to learn unseen cases and aggregate the results. The entire dataset can be stored in a single processor, with equivalent or distinct feature selection methods that allow access to all or portions of it. This strategy known as ensemble learning, has recently received considerable attention [378]. This technique derives is motivated by the fact that, because significant variance is a problem with feature selection methods, one potential solution is to use an ensemble approach by combining methods [379], [380]. Several existing feature selection approaches are unlikely to scale well. However, when dealing with millions of features, they can be redundant. Distributing the data, making feature selections on each split, and combining the results could be one method. In the past decade, several frameworks for distributed learning have been developed. In the last decade, new models for executing distributed learning have been developed, including MapReduce [381], Hadoop [382], Apache Spark [383], and MLib [384]. Another unexplored area of research is the use of graphics processing units (GPUs) to distribute and, accelerate calculations in FS algorithms. The ultimate goal is to use GPU resources to modify the existing state-of-the-art FS methods so that they can handle millions of features quickly and accurately.

E. REAL-TIME PROCESSING

Batch learning algorithms cannot deal with continuously flowing data streams and, require the use of online methods. In recent years, incorporating new data on-demand, online learning [385] has been the practice of rewriting and updating models. It has also become an exciting topic because it acknowledges the critical challenges of activities. The mapping process was monitored in real-time when new samples were received. Because learning data in a sequential manner may be an option for large datasets, online learning may be effective. The same emphasis has not been placed on the selection of online features as it is on online learning. Despite this, a few articles have described strategies for selecting relevant features in a setting that includes both fresh samples and new features. Zhang *et al.* [386] presented an incremental feature subset selection algorithm based on a Boolean matrix that efficiently identifies valuable features for a given data purpose. However, a complete machine learning technique was not used to verify the effectiveness of the FS method. Most online feature selections have been made individually, either by pre-selecting features in a phase separate from the online machine learning phase or by performing an online FS without subsequent online categorization. Consequently, performing real-time analysis and prediction on portable devices for high-dimensional data remains a challenge for artificial intelligence. The challenge is to create dynamic feature selection methods that can change the subset of characteristics chosen when new training instances appear. These techniques should also be implemented in a dynamic feature set that begins empty but fills up as new data are received.

F. EVALUATION MEASURES

One of the main variables in the evolution computation for the FS is the evaluation measure, which generates the fitness function. It has a significant impact on the computing time, classification performance, and search space landscape. For wrapper approaches and many filter approaches, the evaluation procedure consumes the majority of computing time [377], [387], [388]. Although other efficient evaluation measures exist, such as mutual information [389], [390], [391], [392], they only analyze individual features instead of a group of features. Finding complicated feature interaction, on the other hand, is extremely difficult, and only a few studies have been conducted in this area [393]. Rough set-based measures [394], [395] can analyze groups of features [394], [395], [396], [397]; however, they are frequently expensive. Furthermore, numerous studies have shown that filter approaches do not scale well beyond thousands, if not millions of characteristics [364]. As a result, new measures for feature selection are still needed, especially when working with enormous challenges. Multiple distinct solutions to FS challenges may have the same fitness values. A slight (significant) modification in the solution can result in a significant (slight) divergence in the fitness value. This refers to the difficulty level of the challenge. As a result, establishing new measures to smooth the fitness landscape will significantly lower the difficulty of the task and aid in the development of appropriate search algorithms.

G. SPECIFICATION OF HYPER-PARAMETERS

The majority of unsupervised FS methods (filter, wrapper, and hybrids) demand the definition of hyper-parameters such as the set of features, cluster size, and other parameters relevant to the FS methodology utilized by each method. Such knowledge does not exist in reality. It is nearly impossible to determine the ideal parameter values for each dataset. As a result, selecting ideal parameter values automatically is still a work in progress.

H. VISUALIZATION AND INTERPRETABILITY

Several dimensionality reduction strategies for data visualization and preprocessing have been introduced in the last few years. Although the goal may be improved visualization, most solutions have the drawback: that the characteristics being represented are changes in the original features [398], [399], [400]. When model interpretability is crucial, FS is the dimensionality reduction strategy. They performed because the model was only as good as its features. They will continue to play an important role in the model interpretation. Users can choose between the two criteria for the FS and model creation processes. More interactive model visualizations can change the input parameters in response to model challenges and visualize future events. The other is a more interactive feature selection process where they are encouraged to iterate utilizing interactive visualizations. The goal was to make the results more interpretable by allowing user-friendly visualization. The complexities of big data applications highlight the importance of minimizing visual complexity. Although most studies have focused on FS and visualization separately, the display of data features may play an important role in real-world high dimensionality contexts. While visualization tools are constantly used to analyze and make complex data understandable, the quality of the corresponding decision-making is frequently compromised. Because the tools refused to acknowledge the role of heuristics, biases, and other factors in human-computer interaction situations, interactive tools such as those suggested by Krause *et al.* [401] are intriguing research topics.

IX. CONCLUSION

Feature selection is a dimensionality reduction strategy that separates important feature subsets from irrelevant and redundant ones. The importance of FS for data processing has grown significantly with the increease in the number of available FS methods. In addition to well-known FS approaches, this study presents a strategic categorization. Different search strategies, and standard learning methods for improving learning performance are discussed. A good representation of a wide range of algorithms based on the evaluation criteria is also presented. These FS approaches, on the other hand, have gained usability but still have potential. This potentiality is presented systematically, and some challenges in retrieving this potential are also illustrated. In this study, several result validation and performance measurement methodologies were also highlighted to quantify the efficiency and effectiveness of feature selection. In addition, various of application categories have been added to demonstrate the breadth of feature availability.

ACKNOWLEDGMENT

The authors would like to thank the Computer Vision and Pattern Recognition Laboratory, University of Asia Pacific, the Advanced Machine Learning (AML) Laboratory, BUBT, and the Database System Laboratory, UoA, for resource sharing and research collaboration.

REFERENCES

[1] A. J. Miller, "Selection of subsets of regression variables," *J. Roy. Stat. Soc., A (Gen.)*, vol. 147, no. 3, pp. 389–410, 1984.

[2] H. Hotelling, "The selection of variates for use in prediction with some comments on the general problem of nuisance parameters," *Ann. Math. Statist.*, vol. 11, no. 3, pp. 271–283, 1940.

[3] R. R. Hocking, "A biometrics invited paper. The analysis and selection of variables in linear regression," *Biometrics*, vol. 32, no. 1, pp. 1–49, 1976.

[4] V. Kumar and S. Minz, "Feature selection: A literature review," *Smart Comput. Rev.*, vol. 4, no. 3, pp. 211–229, 2014.

[5] F. Tan, "Improving feature selection techniques for machine learning," Georgia Stage Univ., Atlanta, GA, USA, 2007.

[6] N. Mlambo, W. K. Cheruiyot, and M. W. Kimwele, "A survey and comparative study of filter and wrapper feature selection techniques," *Int. J. Eng. Sci.*, vol. 5, no. 8, pp. 57–67, 2016.

[7] C. Shao, K. Paynabar, T. H. Kim, J. J. Jin, and S. J. Hu, "Feature selection for manufacturing process monitoring using cross-validation," *J. Manuf. Syst.*, vol. 32, no. 4, pp. 550–555, Oct. 2013.

[8] L. Ma, M. Li, Y. Gao, T. Chen, X. Ma, and L. Qu, "A novel wrapper approach for feature selection in object-based image classification using polygon-based cross-validation," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 409–413, Mar. 2017.

[9] S. Visa, B. Ramsay, A. L. Ralescu, and E. Van Der Knaap, "Confusion matrix-based feature selection," in *Proc. MAICS*, vol. 710, 2011, pp. 120–127.

[10] I. Kabasakal and H. Soyuer, "A Jaccard similarity-based model to match stakeholders for collaboration in an industry-driven portal," *Proceedings*, vol. 74, no. 1, p. 15, 2021.

[11] N. Kushwaha and M. Pant, "Link based BPSO for feature selection in big data text clustering," *Future Gener. Comput. Syst.*, vol. 82, pp. 190–199, May 2018.

[12] T. Naghibi, S. Hoffmann, and B. Pfister, "A semidefinite programming based search strategy for feature selection with mutual information measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1529–1541, Aug. 2016.

[13] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. Comput.*, vol. C-26, no. 9, pp. 917–922, Sep. 1977.

[14] H. Liu and H. Motoda, *Feature Extraction, Construction and Selection: A Data Mining Perspective*, vol. 453. Norwell, MA, USA: Springer, 1998.

[15] D. J. Stracuzzi, "Randomized feature selection," in *Computational Methods of Feature Selection*. Boca Raton, FL, USA: CRC Press, 2007, pp. 57–78.

[16] J. Bins and B. A. Draper, "Feature selection from huge feature sets," in *Proc. 18th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 159–165.

[17] L. Luo, L. Ye, M. Luo, D. Huang, H. Peng, and F. Yang, "Methods of forward feature selection based on the aggregation of classifiers generated by single attribute," *Comput. Biol. Med.*, vol. 41, no. 7, pp. 435–441, 2011.

[18] S. Déjean, R. T. Ionescu, J. Mothe, and M. Z. Ullah, "Forward and backward feature selection for query performance prediction," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, 2020, pp. 690–697.

[19] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.

[20] J. Doak, "An evaluation of feature selection methods and their application to computer security," Dept. Comput. Sci., Univ. California, USA, Tech. Rep. CSE-92-18, 1992.

[21] W. Zhang, "Branch-and-bound search algorithms and their computational complexity," Marina Del Rey Inf. Sci. Inst., Univ. Southern California, Los Angeles, CA, USA, Tech. Rep. ADA314598, 1996.

[22] Y. Liu, C.-M. Li, H. Jiang, and K. He, "A learning based branch and bound for maximum common subgraph related problems," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 3, pp. 2392–2399.

[23] E. Sadeqi Azer, F. R. Mehrabadi, and S. Malikić, "PhISCS-BnB: A fast branch and bound algorithm for the perfect tumor phylogeny reconstruction problem," *Bioinformatics*, vol. 36, no. 1, pp. i169–i176, 2020.

[24] A. Watanabea, R. Tamurab, Y. Takanod, and R. Miyashiroe, "Branch-and-bound algorithm for optimal sparse canonical correlation analysis," *Benefits*, vol. 13, no. 23, pp. 36–38.

[25] F. Jaeckle, J. Lu, and M. Pawan Kumar, "Neural network branch-and-bound for neural network verification," 2021, *arXiv:2107.12855*.

[26] A. Parjadis, Q. Cappart, L.-M. Rousseau, and D. Bergman, "Improving branch-and-bound using decision diagrams and reinforcement learning," in *Proc. Int. Conf. Integr. Constraint Program., Artif. Intell., Oper. Res.*, in Lecture Notes in Computer Science, vol. 12735. Vienna, Austria: Springer, 2021, pp. 446–455.

[27] P. Gupta, D. Doermann, and D. DeMenthon, "Beam search for feature selection in automatic SVM defect classification," in *Proc. Int. Conf. Pattern Recognit.*, vol. 2, Aug. 2002, pp. 212–215.

[28] D. Ververidis and C. Kotropoulos, "Sequential forward feature selection with low computational cost," in *Proc. 13th Eur. Signal Process. Conf.*, Sep. 2005, pp. 1–4.

[29] A. Marcano-Cedeño, J. Quintanilla-Domínguez, M. Cortina-Januchs, and D. Andina, "Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network," in *Proc. 36th Annu. Conf. IEEE Ind. Electron. Soc.*, Nov. 2010, pp. 2845–2850.

[30] K. Mani and P. Kalpana, "An efficient feature selection based on Bayes theorem, self information and sequential forward selection," *Int. J. Inf. Eng. Electron. Bus.*, vol. 8, no. 6, pp. 46–54, 2016.

[31] S. Shafiee, L. M. Lied, I. Burud, J. A. Dieseth, M. Alsheikh, and M. Lillemo, "Sequential forward selection and support vector regression in comparison to LASSO regression for spring wheat yield prediction based on UAV imagery," *Comput. Electron. Agricult.*, vol. 183, Apr. 2021, Art. no. 106036.

[32] Y. Yulianti and A. Saifudin, "Sequential feature selection in customer churn prediction based on Naive Bayes," in *Proc. IOP Conf., Mater. Sci. Eng.*, 2020, vol. 879, no. 1, Art. no. 012090.

[33] S. J. Reeves and Z. Zhe, "Sequential algorithms for observation selection," *IEEE Trans. Signal Process.*, vol. 47, no. 1, pp. 123–132, Jan. 1999.

[34] R. Aggrawal and S. Pal, "Sequential feature selection and machine learning algorithm-based patient's death events prediction and diagnosis in heart disease," *SN Comput. Sci.*, vol. 1, no. 6, p. 344, 2020.

[35] L. Ji, L. Zhu, L. Wang, Y. Xi, K. Yu, and X. Geng, "FastVGBS: A fast version of the volume-gradient-based band selection method for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 3, pp. 514–517, Mar. 2020.

[36] V. Karunakaran, V. Rajasekar, and S. I. T. Joseph, "Exploring a filter and wrapper feature selection techniques in machine learning," in *Computational Vision and Bio-Inspired Computing*, vol. 1318. Singapore: Springer, 2021, pp. 497–506.

[37] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.

[38] P. Somol, J. Novovičová, and P. Pudil, "Efficient feature subset selection and subset size optimization," in *Pattern Recognition Recent Advances*, vol. 1. Croatia: Books on Demand, 2010.

[39] M. Savadkoohi, T. Oladunni, and L. Thompson, "A machine learning approach to epileptic seizure prediction using electroencephalogram (EEG) Signal," *Biocybern. Biomed. Eng.*, vol. 40, no. 3, pp. 1328–1341, 2020.

[40] X. Peng, K. Cheng, J. Lang, Z. Zhang, T. Cai, and S. Duan, "Short-term wind power prediction for wind farm clusters based on SFFS feature selection and BLSTM deep learning," *Energies*, vol. 14, no. 7, p. 1894, 2021.

[41] T. Desyani, A. Saifudin, and Y. Yulianti, "Feature selection based on Naive Bayes for caesarean section prediction," in *Proc. IOP Conf., Mater. Sci. Eng.*, 2020, vol. 879, no. 1, Art. no. 012091.

[42] W. Dai, Y. Fang, and B. Hu, "Feature selection in interactive face retrieval," in *Proc. 4th Int. Congr. Image Signal Process.*, vol. 3, Oct. 2011, pp. 1358–1362.

[43] K. Bouzoubaa, Y. Taher, and B. Nsiri, "Dos attack forecasting: A comparative study on wrapper feature selection," in *Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, Jun. 2020, pp. 1–7.

[44] S. Rose, S. Nickolas, and S. Sangeetha, "A recursive ensemble-based feature selection for multi-output models to discover patterns among the soil nutrients," *Chemometrics Intell. Lab. Syst.*, vol. 208, Jan. 2021, Art. no. 104221.

[45] L. Chen, M. Wu, W. Pedrycz, and K. Hirota, "AdaBoost-KNN with direct optimization for dynamic emotion recognition," in *Emotion Recognition and Understanding for Emotional Human-Robot Interaction Systems*, vol. 926. Berlin, Germany: Springer, 2021, pp. 41–55.

[46] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.

[47] C. Vandana and A. A. Chikkamannur, "Feature selection: An empirical study," *Int. J. Eng. Trends Technol.*, vol. 69, no. 2, pp. 165–170, 2021.

[48] H. Liu, J. Wang, J. Gao, S. Liu, X. Liu, Z. Zhao, D. Guo, and G. Dan, "A comprehensive hierarchical classification based on multi-features of breast DCE-MRI for cancer diagnosis," *Med. Biol. Eng. Comput.*, vol. 58, no. 10, pp. 2413–2425, 2020.

[49] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[50] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm," *Int. J. Mach. Learn. Cybern.*, vol. 11, pp. 1501–1529, 2020.

[51] P. Civicioglu, "Transforming geocentric Cartesian coordinates to geodetic coordinates by using differential search algorithm," *Comput. Geosci.*, vol. 46, pp. 229–247, Sep. 2012.

[52] H. Salimi, "Stochastic fractal search: A powerful metaheuristic algorithm," *Knowl.-Based Syst.*, vol. 75, pp. 1–18, Feb. 2015.

[53] P. Civicioglu, "Backtracking search optimization algorithm for numerical optimization problems," *Appl. Math. Comput.*, vol. 219, no. 15, pp. 8121–8144, Apr. 2013.

[54] T. Dhivyaprabha, P. Subashini, and M. Krishnaveni, "Synergistic fibroblast optimization: A novel nature-inspired computing algorithm," *Frontiers Inf. Technol. Electron. Eng.*, vol. 19, no. 7, pp. 815–833, 2018.

[55] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

[56] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.

[57] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: Charged system search," *Acta Mech.*, vol. 213, nos. 3–4, pp. 267–289, Sep. 2010.

[58] H. Shah-Osseini, "Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation," *Int. J. Comput. Sci. Eng.*, vol. 6, nos. 1–2, pp. 132–140, 2011.

[59] E. Cuevas, D. Oliva, D. Zaldivar, M. Pérez-Cisneros, and H. Sossa, "Circle detection using electro-magnetism optimization," *Inf. Sci.*, vol. 182, no. 1, pp. 40–55, 2012.

[60] K. Tamura and K. Yasuda, "Spiral optimization—A new multipoint search method," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2011, pp. 1759–1764.

[61] F. Farrahi Moghaddam, R. Farrahi Moghaddam, and M. Cheriet, "Curved space optimization: A random search based on general relativity theory," 2012, *arXiv:1208.2214*.

[62] A. Kaveh and M. Khayatazad, "A new meta-heuristic method: Ray optimization," *Comput. Struct.*, vol. 112, pp. 283–294, Dec. 2012.

[63] M. Abdechiri, M. R. Meybodi, and H. Bahrami, "Gases Brownian motion optimization: An algorithm for optimization (GBMO)," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2932–2946, 2013.

[64] S. Moein and R. Logeswaran, "KGMO: A swarm optimization algorithm based on the kinetic energy of gas molecules," *Inf. Sci.*, vol. 275, pp. 127–144, Aug. 2014.

[65] A. Kaveh and V. R. Mahdavi, "Colliding bodies optimization: A novel meta-heuristic method," *Comput. Struct.*, vol. 139, pp. 18–27, Jan. 2014.

[66] A. Kaveh and T. Bakhshpoori, "Water evaporation optimization: A novel physically inspired optimization algorithm," *Comput. Struct.*, vol. 167, pp. 69–85, Apr. 2016.

[67] A. Kaveh and A. Dadras, "A novel meta-heuristic optimization algorithm: Thermal exchange optimization," *Adv. Eng. Softw.*, vol. 110, pp. 69–84, Aug. 2017.

[68] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Inf. Sci.*, vol. 222, pp. 175–184, Feb. 2012.

[69] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems," *Comput. Struct.*, vols. 110–111, pp. 151–166, Nov. 2012.

[70] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2592–2612, May 2013.

[71] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.

[72] A. Tabari and A. Ahmad, "A new optimization method: Electro-search algorithm," *Comput. Chem. Eng.*, vol. 103, pp. 1–11, Aug. 2017.

[73] A. H. Kashan, "League championship algorithm: A new algorithm for numerical function optimization," in *Proc. Int. Conf. Soft Comput. Pattern Recognit.*, Dec. 2009, pp. 43–48.

[74] N. Ghorbani and E. Babaei, "Exchange market algorithm," *Appl. Soft Comput.*, vol. 19, pp. 177–187, Jun. 2014.

[75] Y. Xu, Z. Cui, and J. Zeng, "Social emotional optimization algorithm for nonlinear constrained optimization problems," in *Proc. Int. Conf. Swarm, Evol., Memetic Comput.*, in Lecture Notes in Computer Science, vol. 6466. Berlin, Germany: Springer, 2010, pp. 583–590.

[76] Y. Shi, "Brain storm optimization algorithm," in *Proc. Int. Conf. Swarm Intell.*, in Lecture Notes in Computer Science, vol. 6728. Berlin, Germany: Springer, 2011, pp. 303–309.

[77] R. V. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *Int. J. Ind. Eng. Comput.*, vol. 7, no. 1, pp. 19–34, 2016.

[78] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, nos. 1–4, pp. 131–156, 1997.

[79] A. G. Karegowda, M. A. Jayaram, and A. S. Manjunath, "Feature subset selection problem using wrapper approach in supervised learning," *Int. J. Comput. Appl.*, vol. 1, no. 7, pp. 13–17, Feb. 2010.

[80] A. G. Karegowda, M. A. Jayaram, and A. S. Manjunath, "Feature subset selection using cascaded GA & CFS: A filter approach in supervised learning," *Int. J. Comput. Appl.*, vol. 23, no. 2, pp. 1–10, 2011.

[81] J. Frank, "Artificial intelligence and intrusion detection: Current and future directions," in *Proc. 17th Nat. Comput. Secur. Conf.*, Baltimore, MD, USA, vol. 10, 1994, pp. 1–12.

[82] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated Annealing: Theory and Applications*. Springer, 1987, pp. 7–15.

[83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[84] F. S. Hosseini, B. Choubin, A. Mosavi, N. Nabipour, S. Shamshirband, H. Darabi, and A. T. Haghighi, "Flash-flood hazard assessment using ensembles and Bayesian-based machine learning models: Application of the simulated annealing feature selection method," *Sci. Total Environ.*, vol. 711, Apr. 2020, Art. no. 135161.

[85] M. Abdel-Basset, W. Ding, and D. El-Shahat, "A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection," *Artif. Intell. Rev.*, vol. 54, no. 1, pp. 593–637, 2021.

[86] A. C. Pandey and D. S. Rajpoot, "Feature selection method based on grey wolf optimization and simulated annealing," *Recent Adv. Comput. Sci. Commun. (Formerly: Recent Patents Comput. Sci.)*, vol. 14, no. 2, pp. 635–646, 2021.

[87] Z. M. Elgamal, N. B. M. Yasin, M. Tubishat, M. Alswaitti, and S. Mirjalili, "An improved Harris Hawks optimization algorithm with simulated annealing for feature selection in the medical field," *IEEE Access*, vol. 8, pp. 186638–186652, 2020.

[88] B. Selman and C. P. Gomes, "Hill-climbing search," *Encyclopedia Cogn. Sci.*, vol. 81, p. 82, Jan. 2006.

[89] M. E. Farmer, S. Bapna, and A. K. Jain, "Large scale feature selection using modified random mutation Hill climbing," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 2, Aug. 2004, pp. 287–290.

[90] S. Goswami, S. Chakraborty, P. Guha, A. Tarafdar, and A. Kedia, "Filter-based feature selection methods using Hill climbing approach," in *Natural Computing for Unsupervised Learning*. Springer, 2019, pp. 213–234.

[91] M. Ghosh, T. Kundu, D. Ghosh, and R. Sarkar, "Feature selection for facial emotion recognition using late hill-climbing based memetic algorithm," *Multimedia Tools Appl.*, vol. 78, no. 18, pp. 25753–25779, 2019.

[92] K. K. Ghosh, S. Ahmed, P. K. Singh, Z. W. Geem, and R. Sarkar, "Improved binary sailfish optimizer based on adaptive β-Hill climbing for feature selection," *IEEE Access*, vol. 8, pp. 83548–83560, 2020.

[93] B. Chatterjee, T. Bhattacharyya, K. K. Ghosh, P. K. Singh, Z. W. Geem, and R. Sarkar, "Late acceptance Hill climbing based social ski driver algorithm for feature selection," *IEEE Access*, vol. 8, pp. 75393–75408, 2020.

[94] M. Nekkaa and D. Boughaci, "A memetic algorithm with support vector machine for feature selection and classification," *Memetic Comput.*, vol. 7, no. 1, pp. 59–73, 2015.

[95] C.-C. Lin, J.-R. Kang, Y.-L. Liang, and C.-C. Kuo, "Simultaneous feature and instance selection in big noisy data using memetic variable neighborhood search," *Appl. Soft Comput.*, vol. 112, Nov. 2021, Art. no. 107855.

[96] N. Chakraborty, A. Ray, A. F. Mollah, S. Basu, and R. Sarkar, "A framework for multi-lingual scene text detection using k-means++ and memetic algorithms," in *Machine Learning for Intelligent Multimedia Analytics* (Studies in Big Data), vol. 82. Singapore: Springer, 2021, pp. 167–187.

[97] G. Acampora, V. Cataudella, P. R. Hegde, P. Lucignano, G. Passarelli, and A. Vitiello, "Memetic algorithms for mapping p-body interacting systems in effective quantum 2-body Hamiltonians," *Appl. Soft Comput.*, vol. 110, Oct. 2021, Art. no. 107634.

[98] H. Liu and R. Setiono, "A probabilistic approach to feature selection—A filter solution," in *Proc. ICML*, vol. 96, 1996, pp. 319–327.

[99] S. Kashef, H. Nezamabadi-Pour, and B. Nikpour, "Multilabel feature selection: A comprehensive review and guiding experiments," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 8, no. 2, p. e1240, 2018.

[100] K. V. Price, "Differential evolution: A fast and simple numerical optimizer," in *Proc. North Amer. Fuzzy Inf. Process.*, Jun. 1996, pp. 524–527.

[101] M. Georgioudakis and V. Plevris, "A comparative study of differential evolution variants in constrained structural optimization," *Frontiers Built Environ.*, vol. 6, p. 102, Jul. 2020.

[102] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw. Proc.*, vol. 1, pp. 33–57, Aug. 2007.

[103] A. A. de Moura Meneses, M. D. Machado, and R. Schirru, "Particle swarm optimization applied to the nuclear reload problem of a pressurized water reactor," *Prog. Nucl. Energy*, vol. 51, no. 2, pp. 319–326, 2009.

[104] Wikipedia Contributors. (2021). *Particle Swarm Optimization—Wikipedia, the Free Encyclopedia*. Accessed: Nov. 15, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Particle_swarm_optimization&oldid=1050772675

[105] K. Y. Lee and J.-B. Park, "Application of particle swarm optimization to economic dispatch problem: Advantages and disadvantages," in *Proc. IEEE PES Power Syst. Conf. Expo.*, Oct./Nov. 2006, pp. 188–192.

[106] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[107] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," in *Handbook of Pattern Recognition and Computer Vision*. Singapore: World Scientific, 1993, pp. 88–107.

[108] R. Jagdhuber, M. Lang, A. Stenzl, J. Neuhaus, and J. Rahnenführer, "Cost-Constrained feature selection in binary classification: Adaptations for greedy forward selection and genetic algorithms," *BMC Bioinf.*, vol. 21, no. 1, pp. 1–21, 2020.

[109] F. Amini and G. Hu, "A two-layer feature selection method using genetic algorithm and elastic net," *Expert Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 114072.

[110] A.-D. Li, B. Xue, and M. Zhang, "Multi-objective feature selection using hybridization of a genetic algorithm and direct multisearch for key quality characteristic selection," *Inf. Sci.*, vol. 523, pp. 245–265, Jun. 2020.

[111] R. Guha, M. Ghosh, S. Kapri, S. Shaw, S. Mutsuddi, V. Bhateja, and R. Sarkar, "Deluge based genetic algorithm for feature selection," *Evol. Intell.*, vol. 14, no. 2, pp. 357–367, 2021.

[112] L. Abualigah and A. J. Dulaimi, "A novel feature selection method for data mining tasks using hybrid sine cosine algorithm and genetic algorithm," *Cluster Comput.*, vol. 21, pp. 2161–2176, Feb. 2021.

[113] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, Jun. 2004, Art. no. 066138.

[114] X.-F. Song, Y. Zhang, D.-W. Gong, and X.-Y. Sun, "Feature selection using bare-bones particle swarm optimization with mutual information," *Pattern Recognit.*, vol. 112, Apr. 2021, Art. no. 107804.

[115] B. A. S. Al-Rimy, M. A. Maarof, M. Alazab, S. Z. M. Shaid, F. A. Ghaleb, A. Almalawi, A. M. Ali, and T. Al-Hadhrami, "Redundancy coefficient gradual up-weighting-based mutual information feature selection technique for crypto-ransomware early detection," *Future Gener. Comput. Syst.*, vol. 115, pp. 641–658, Feb. 2021.

[116] Z.-C. Sha, Z.-M. Liu, C. Ma, and J. Chen, "Feature selection for multi-label classification by maximizing full-dimensional conditional mutual information," *Appl. Intell.*, vol. 51, no. 1, pp. 326–340, Aug. 2021.

[117] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing* (Springer Topics in Signal Processing), vol. 2. Berlin, Germany, 2009, pp. 1–4.

[118] H. Zhou, Z. Deng, Y. Xia, and M. Fu, "A new sampling method in particle filter based on Pearson correlation coefficient," *Neurocomputing*, vol. 216, pp. 208–215, Dec. 2016.

[119] M. A. Hall, "Correlation-based feature selection for machine learning," Univ. Waikato, Hamilton, New Zealand, 1999.

[120] Y. Piao, M. Piao, K. Park, and K. H. Ryu, "An ensemble correlation-based gene selection algorithm for cancer classification with gene expression data," *Bioinformatics*, vol. 28, no. 24, pp. 3306–3315, 2012.

[121] Z. Chen, Z. Hong, and D. Tao, "An experimental survey on correlation filter-based tracking," 2015, *arXiv:1509.05520*.

[122] A. Gelbukh, *Computational Linguistics and Intelligent Text Processing, in 14th International Conference on Intelligent Text Processing and Computational Linguistics CICLing 2013*, vol. 7816. Greece: Springer, 2013.

[123] H. Uğuz, "A hybrid system based on information gain and principal component analysis for the classification of transcranial Doppler signals," *Comput. Methods Programs Biomed.*, vol. 107, no. 3, pp. 598–609, 2012.

[124] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, no. 10, pp. 1205–1224, 2004.

[125] Z. A. Baig, S. M. Sait, and A. Shaheen, "GMDH-based networks for intelligent intrusion detection," *Eng. Appl. Artif. Intell.*, vol. 26, no. 7, pp. 1731–1740, Aug. 2013.

[126] H. Ezzat Ibrahim, S. M. Badr, and M. A. Shaheen, "Adaptive layered approach using machine learning techniques with gain ratio for intrusion detection systems," 2012, *arXiv:1210.7650*.

[127] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 18, 2005, pp. 1–8.

[128] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A survey on semi-supervised feature selection methods," *Pattern Recognit.*, vol. 64, pp. 141–158, Apr. 2016.

[129] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," 2012, *arXiv:1202.3725*.

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

[131] M. Gan and L. Zhang, "Q-learning with Fisher score for feature selection of large-scale data sets," in *Proc. Int. Conf. Knowl. Sci., Eng. Manage.*, in Lecture Notes in Computer Science, vol. 12816. China: Springer, 2021, pp. 306–318.

[132] J. Zhang, D. Xu, K. Hao, Y. Zhang, W. Chen, J. Liu, R. Gao, C. Wu, and Y. De Marinis, "FS–GBDT: Identification multicancer-risk module via a feature selection algorithm by integrating Fisher score and GBDT," *Briefings Bioinf.*, vol. 22, no. 3, 2021, Art. no. bbaa189.

[133] Y. Park and W. Chung, "Optimal channel selection using correlation coefficient for CSP based EEG classification," *IEEE Access*, vol. 8, pp. 111514–111521, 2020.

[134] K. L. Devi, P. Subathra, and P. Kumar, "Tweet sentiment classification using an ensemble of machine learning supervised classifiers employing statistical feature selection methods," in *Proc. 5th Int. Conf. Fuzzy Neuro Comput. (FANCCO)*. Springer, 2015, pp. 1–13.

[135] H. G. Schulze, R. B. Foist, A. Ivanov, and R. F. Turner, "Chi-squared-based filters for high-fidelity signal-to-noise ratio enhancement of spectra," *Appl. Spectrosc.*, vol. 62, no. 8, pp. 847–853, 2008.

[136] A.-M. Bidgoli and M. N. Parsa, "A hybrid feature selection by resampling, chi squared and consistency evaluation techniques," *World Acad. Sci., Eng. Technol.*, vol. 68, pp. 276–285, Aug. 2012.

[137] X.-Q. Zeng, G.-Z. Li, and S.-F. Chen, "Gene selection by using an improved fast correlation-based filter," in *Proc. IEEE Int. Conf. Bioinf. Biomed. Workshops (BIBMW)*, Dec. 2010, pp. 625–630.

[138] D. Zhang, S. Chen, and Z.-H. Zhou, "Constraint score: A new filter method for feature selection with pairwise constraints," *Pattern Recognit.*, vol. 41, no. 5, pp. 1440–1451, 2008.

[139] V. Kumar, C. Roche, S. Overman, R. Simovitch, P.-H. Flurin, T. Wright, J. Zuckerman, H. Routman, and A. Teredesai, "Using machine learning to predict clinical outcomes after shoulder arthroplasty with a minimal feature set," *J. Shoulder Elbow Surg.*, vol. 30, no. 5, pp. e225–e236, 2021.

[140] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning Proceedings*. Amsterdam, The Netherlands: Elsevier, 1992, pp. 249–256.

[141] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Proc. Eur. Conf. Mach. Learn.* Springer, 1994, pp. 171–182.

[142] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Mach. Learn.*, vol. 53, nos. 1–2, pp. 23–69, Oct. 2003.

[143] J. Yang, Z. Zhu, S. He, and Z. Ji, "Minimal-redundancy-maximal-relevance feature selection using different relevance measures for omics data classification," in *Proc. IEEE Symp. Comput. Intell. Bioinf. Comput. Biol. (CIBCB)*, Apr. 2013, pp. 246–251.

[144] X. Gu, J. Guo, L. Xiao, and C. Li, "Conditional mutual information-based feature selection algorithm for maximal relevance minimal redundancy," *Appl. Intell.*, vol. 52, pp. 1436–1447, May 2021.

[145] X. Gu, J. Guo, L. Xiao, T. Ming, and C. Li, "A feature selection algorithm based on equal interval division and minimal-redundancy–maximal-relevance," *Neural Process. Lett.*, vol. 51, no. 2, pp. 1237–1263, Nov. 2020.

[146] S. Alelyani, *On Feature Selection Stability: A Data Perspective*. Tucson, AZ, USA: Arizona State Univ., 2013.

[147] P. Padungweang, C. Lursinsap, and K. Sunat, "Univariate filter technique for unsupervised feature selection using a new Laplacian score based local nearest neighbors," in *Proc. Asia–Pacific Conf. Inf. Process.*, vol. 2, Jul. 2009, pp. 196–200.

[148] D. Guru, M. Suhil, L. N. Raju, and N. V. Kumar, "An alternative framework for univariate filter based feature selection for text categorization," *Pattern Recognit. Lett.*, vol. 103, pp. 23–31, Feb. 2018.

[149] J. Hua, W. D. Tembe, and E. R. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data," *Pattern Recognit.*, vol. 42, no. 3, pp. 409–424, 2009.

[150] C. Lai, M. J. T. Reinders, and L. Wessels, "Random subspace method for multivariate feature selection," *Pattern Recognit. Lett.*, vol. 27, no. 10, pp. 1067–1076, 2006.

[151] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, Oct. 2007.

[152] V. Trevino and F. Falciani, "GALGO: An R package for multivariate variable selection using genetic algorithms," *Bioinformatics*, vol. 22, pp. 1154–1156, May 2006.

[153] N. Sánchez-Marono, A. Alonso-Betanzos, and M. Tombilla-Sanromán, "Filter methods for feature selection—A comparative study," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.*, in Lecture Notes in Computer Science, vol. 4881. Berlin, Germany: Springer, 2007, pp. 178–187.

[154] M. Dash, K. Choi, P. Scheuermann, and H. Liu, "Feature selection for clustering—A filter solution," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2002, pp. 115–122.

[155] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "On the effectiveness of discretization on gene selection of microarray data," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2010, pp. 1–8.

[156] N. Sánchez-Marono, A. Alonso-Betanzos, P. García-González, and V. Bolón-Canedo, "Multiclass classifiers vs multiple binary classifiers using filters for feature selection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2010, pp. 1–8.

[157] A. J. Ferreira and M. A. T. Figueiredo, "An unsupervised approach to feature discretization and selection," *Pattern Recognit.*, vol. 45, no. 9, pp. 3048–3060, Sep. 2012.

[158] F. F. G. Navarro and L. A. B. Munoz, "Gene subset selection in microarray data using entropic filtering for cancer classification," *Expert Syst.*, vol. 26, no. 1, pp. 113–124, 2009.

[159] V. Bolón-Canedo, S. Seth, N. Sánchez-Marono, A. Alonso-Betanzos, and J. C. Príncipe, "Statistical dependence measure for feature selection in microarray datasets," in *Proc. ESANN*, 2011.

[160] S. Student and K. Fujarewicz, "Stable feature selection and classification algorithms for multiclass microarray data," *Biol. Direct*, vol. 7, no. 1, pp. 1–20, 2012.

[161] Y. Liu and Y. F. Zheng, "FS_SFS: A novel feature selection method for support vector machines," *Pattern Recognit.*, vol. 39, no. 7, pp. 1333–1345, 2006.

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

[163] X.-W. Chen and J. C. Jeong, "Enhanced recursive feature elimination," in *Proc. 6th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2007, pp. 429–435.

[164] K. Yan and D. Zhang, "Feature selection and analysis on correlated gas sensor data with recursive feature elimination," *Sens. Actuators B, Chem.*, vol. 212, pp. 353–363, Jun. 2015.

[165] B. Richhariya, M. Tanveer, and A. H. Rashid, "Diagnosis of Alzheimer's disease using universum support vector machine based recursive feature elimination (USVM-RFE)," *Biomed. Signal Process. Control*, vol. 59, May 2020, Art. no. 101903.

[166] Q. Chen, Z. Meng, and R. Su, "WERFE: A gene selection algorithm based on recursive feature elimination and ensemble strategy," *Frontiers Bioeng. Biotechnol.*, vol. 8, p. 496, May 2020.

[167] N. Rtayli and N. Enneya, "Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization," *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102596.

[168] P. Romanski and L. Kotthoff, "Fselector: Selecting attributes," in *Vienna: R Foundation for Statistical Computing*, Vienna, Austria, 2009.

[169] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[170] H. Ahmadpour, O. Bazrafshan, E. Rafiei-Sardooi, H. Zamani, and T. Panagopoulos, "Gully erosion susceptibility assessment in the kondoran watershed using machine learning algorithms and the Boruta feature selection," *Sustainability*, vol. 13, no. 18, p. 10110, 2021.

[171] Z. Ebrahimi-Khusfi, A. R. Nafarzadegan, and F. Dargahian, "Predicting the number of dusty days around the desert wetlands in southeastern Iran using feature selection and machine learning techniques," *Ecol. Indicators*, vol. 125, Jun. 2021, Art. no. 107499.

[172] M. A. Sarder, M. Maniruzzaman, and B. Ahammed, "Feature selection and classification of leukemia cancer using machine learning techniques," *Mach. Learn. Res.*, vol. 5, no. 2, p. 18, 2020.

[173] R. Tang and X. Zhang, "CART decision tree combined with Boruta feature selection for medical data classification," in *Proc. 5th IEEE Int. Conf. Big Data Anal. (ICBDA)*, May 2020, pp. 80–84.

[174] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in *Proc. ICML*, vol. 1, 2001, pp. 74–81.

[175] R. Kohavi and D. Sommerfield, "Feature subset selection using the wrapper method: Overfitting and dynamic search space topology," in *Proc. KDD*, 1995, pp. 192–197.

[176] S. Maldonado and R. Weber, "A wrapper method for feature selection using support vector machines," *Inf. Sci.*, vol. 179, no. 13, pp. 2208–2217, 2009.

[177] V. F. Rodriguez-Galiano, J. A. Luque-Espinar, M. Chica-Olmo, and M. P. Mendes, "Feature selection approaches for predictive modelling of groundwater nitrate pollution: An evaluation of filters, embedded and wrapper methods," *Sci. Total Environ.*, vol. 624, pp. 661–672, May 2018.

[178] R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz, "Incremental wrapper-based gene selection from microarray data for cancer classification," *Pattern Recognit.*, vol. 39, no. 12, pp. 2383–2392, Dec. 2006.

[179] P. Bermejo, J. A. Gámez, and J. M. Puerta, "Speeding up incremental wrapper feature subset selection with Naive Bayes classifier," *Knowl.-Based Syst.*, vol. 55, pp. 140–147, Jan. 2014.

[180] H. Li, C.-J. Li, X.-J. Wu, and J. Sun, "Statistics-based wrapper for feature selection: An implementation on financial distress identification with support vector machine," *Appl. Soft Comput.*, vol. 19, pp. 57–67, Jun. 2014.

[181] M. Alzaqebah, N. Alrefai, E. A. Ahmed, S. Jawarneh, and M. K. Alsmadi, "Neighborhood search methods with moth optimization algorithm as a wrapper method for feature selection problems," *Int. J. Elect. Comput. Eng.*, vol. 10, no. 4, p. 3672, 2020.

[182] T. N. Lal, O. Chapelle, J. Weston, and A. Elisseeff, "Embedded methods," in *Feature Extraction* (Studies in Fuzziness and Soft Computing), vol. 207. Berlin, Germany: Springer, 2006, pp. 137–165.

[183] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 3, pp. 703–715, May 2019.

[184] C.-W. Chen, Y.-H. Tsai, F.-R. Chang, and W.-C. Lin, "Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results," *Expert Syst.*, vol. 37, no. 5, 2020, Art. no. e12553.

[185] Y. Hua, "An efficient traffic classification scheme using embedded feature selection and lightGBM," in *Proc. Inf. Commun. Technol. Conf. (ICTC)*, May 2020, pp. 125–130.

[186] C.-F. Tsai and Y.-T. Sung, "Ensemble feature selection in high dimension, low sample size datasets: Parallel and serial combination approaches," *Knowl.-Based Syst.*, vol. 203, Sep. 2020, Art. no. 106097.

[187] H. Polat, O. Polat, and A. Cetin, "Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models," *Sustainability*, vol. 12, no. 3, p. 1035, 2020.

[188] J. M. Valente and S. Maldonado, "SVR-FFS: A novel forward feature selection approach for high-frequency time series forecasting using support vector regression," *Expert Syst. Appl.*, vol. 160, Dec. 2020, Art. no. 113729.

[189] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.

[190] V. Fonti and E. Belitser, "Feature selection using lasso," *VU Amsterdam Res. Paper Bus. Anal.*, vol. 30, pp. 1–25, Mar. 2017.

[191] R. Muthukrishnan and R. Rohini, "LASSO: A feature selection technique in predictive modeling for machine learning," in *Proc. IEEE Int. Conf. Adv. Comput. Appl. (ICACA)*, Oct. 2016, pp. 18–20.

[192] S. Maldonado, R. Weber, and J. Basak, "Simultaneous feature selection and classification using kernel-penalized support vector machines," *Inf. Sci.*, vol. 181, no. 1, pp. 115–128, 2011.

[193] G. Roffo, S. Melzi, and M. Cristani, "Infinite feature selection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4202–4210.

[194] A.-C. Haury, P. Gestraud, and J.-P. Vert, "The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures," *PLoS ONE*, vol. 6, no. 12, 2011, Art. no. e28210.

[195] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *J. Bio. Comput. Biol.*, vol. 3, no. 2, pp. 185–205, 2005.

[196] P. A. Mundra and J. C. Rajapakse, "SVM-RFE with MRMR filter for gene selection," *IEEE Trans. Nanobiosci.*, vol. 9, no. 1, pp. 31–37, Mar. 2010.

[197] Z. Xiao, E. Dellandrea, W. Dou, and L. Chen, "ESFS: A new embedded feature selection method based on SFS," Ph.D. dissertation, Ecole Centrale Lyon, Université de Lyon, Lyon, France, 2008.

[198] S. Wang, J. Tang, and H. Liu, "Embedded unsupervised feature selection," in *Proc. AAAI Conf. Artif. Intell.*, 2015, vol. 29, no. 1, pp. 1–7.

[199] S. Maldonado and J. López, "Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification," *Appl. Soft Comput.* vol. 67, pp. 94–105, Jun. 2018.

[200] X. Zhang, G. Wu, Z. Dong, and C. Crawford, "Embedded feature-selection support vector machine for driving pattern recognition," *J. Franklin Inst.*, vol. 352, no. 2, pp. 669–685, Feb. 2015.

[201] M. Boussouf, "A hybrid approach to feature selection," in *Proc. Eur. Symp. Princ. Data Mining Knowl. Discovery*, in Lecture Notes in Computer Science, vol. 1510. Berlin, Germany: Springer, 1998, pp. 230–238.

[202] J. Liu and G. Wang, "A hybrid feature selection method for data sets of thousands of variables," in *Proc. 2nd Int. Conf. Adv. Comput. Control*, vol. 2, Mar. 2010, pp. 288–291.

[203] M. Kang, M. R. Islam, J. Kim, J. M. Kim, and M. Pecht, "A hybrid feature selection scheme for reducing diagnostic performance deterioration caused by outliers in data-driven diagnostics," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3299–3310, May 2016.

[204] R. Islam, S. A. Khan, and J.-M. Kim, "Discriminant feature distribution analysis-based hybrid feature selection for online bearing fault diagnosis in induction motors," *J. Sensors*, vol. 2016, Dec. 2016, Art. no. 7145715.

[205] S. Ahmed, M. Zhang, and L. Peng, "Enhanced feature selection for biomarker discovery in LC-MS data using GP," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 584–591.

[206] R. Islam *et al.*, "Maximum class separability based discriminant feature selection using a GA for reliable fault diagnosis of induction motors," in *Proc. Int. Conf. Intell. Comput.*, in Lecture Notes in Computer Science, vol. 9227. China: Springer, 2015, pp. 526–537.

[207] N. Ahmed, J. I. Rafiq, and M. R. Islam, "Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model," *Sensors*, vol. 20, no. 1, p. 317, 2020.

[208] Y. Peng, Z. Wu, and J. Jiang, "A novel feature selection approach for biomedical data classification," *J. Biomed. Inform.*, vol. 43, no. 1, pp. 15–23, Feb. 2010.

[209] R. Alzubi, N. Ramzan, H. Alzoubi, and A. Amira, "A hybrid feature selection method for complex diseases SNPs," *IEEE Access*, vol. 6, pp. 1292–1301, 2018.

[210] C.-P. Lee and Y. Leu, "A novel hybrid feature selection method for microarray data analysis," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 208–213, Jan. 2011.

[211] L.-Y. Chuang, C.-H. Yang, K.-C. Wu, and C.-H. Yang, "A hybrid feature selection method for DNA microarray data," *Comput. Biol. Med.*, vol. 41, no. 4, pp. 228–237, Apr. 2011.

[212] M.-C. Lee, "Using support vector machine with a hybrid feature selection method to the stock trend prediction," *Expert Syst. Appl.*, vol. 36, no. 8, pp. 10896–10904, 2009.

[213] H.-H. Hsu, C.-W. Hsieh, and M.-D. Lu, "Hybrid feature selection by combining filters and wrappers," *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8144–8150, 2011.

[214] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *Proc. Int. Conf. Algorithmic Learn. Theory*, in Lecture Notes in Computer Science, vol. 3734. Berlin, Germany: Springer, 2005, pp. 63–77.

[215] A. Gretton, K. Fukumizu, C. Teo, L. Song, B. Schölkopf, and A. Smola, "A kernel statistical test of independence," in *Proc. NIPS*, vol. 20, 2007, pp. 585–592.

[216] H. Shen, S. Jegelka, and A. Gretton, "Fast kernel-based independent component analysis," *IEEE Trans. Signal Process.*, vol. 57, no. 9, pp. 3498–3511, Sep. 2009.

[217] N. Quadrianto, A. J. Smola, L. Song, and T. Tuytelaars, "Kernelized sorting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1809–1821, Oct. 2010.

[218] M. J. Gangeh, A. Ghodsi, and M. S. Kamel, "Kernelized supervised dictionary learning," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4753–4767, Oct. 2013.

[219] M. J. Gangeh, P. Fewzee, A. Ghodsi, M. S. Kamel, and F. Karray, "Multiview supervised dictionary learning in speech emotion recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 6, pp. 1056–1068, Jun. 2014.

[220] B. Chang, U. Kruger, R. Kustra, and J. Zhang, "Canonical correlation analysis based on Hilbert–Schmidt independence criterion and centered kernel target alignment," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 316–324.

[221] M. J. Gangeh, H. Zarkoob, and A. Ghodsi, "Fast and scalable feature selection for gene expression data using Hilbert–Schmidt independence criterion," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 167–181, Jan./Feb. 2017.

[222] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 907–948, 2020.

[223] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *J. Mach. Learn. Res.*, vol. 5, pp. 845–889, Jan. 2004.

[224] M. Breaban and H. Luchian, "A unifying criterion for unsupervised clustering and feature selection," *Pattern Recognit.*, vol. 44, no. 4, pp. 854–865, 2011.

[225] E. R. Hruschka and T. F. Covoes, "Feature selection for cluster analysis: An approach based on the simplified Silhouette criterion," in *Proc. Int. Conf. Comput. Intell. Modelling, Control Automat. Int. Conf. Intell. Agents, Web Technol. Internet Commerce (CIMCA-IAWTIC)*, vol. 1, Nov. 2005, pp. 32–38.

[226] Y. Kim, W. N. Street, and F. Menczer, "Evolutionary model selection in unsupervised learning," *Intell. Data Anal.*, vol. 6, no. 6, pp. 531–556, 2002.

[227] D. Dutta, P. Dutta, and J. Sil, "Simultaneous feature selection and clustering with mixed features by multi objective genetic algorithm," *Int. J. Hybrid Intell. Syst.*, vol. 11, no. 1, pp. 41–54, 2014.

[228] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1154–1166, Sep. 2004.

[229] V. Roth and T. Lange, "Feature selection in clustering problems," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16, 2003, pp. 473–480.

[230] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 381–396, Mar. 2002.

[231] L. Zuo, L. Li, and C. Chen, "The graph based semi-supervised algorithm with ℓ^1-regularizer," *Neurocomputing*, vol. 149, pp. 966–974, Feb. 2015.

[232] K. Zhang, L. Lan, J. Kwok, S. Vucetic, and B. Parvin, "Scaling up graph-based semisupervised learning via prototype vector machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 444–457, Mar. 2015.

[233] N. N. Pise and P. Kulkarni, "A survey of semi-supervised learning methods," in *Proc. Int. Conf. Comput. Intell. Secur.*, vol. 2, Dec. 2008, pp. 30–34.

[234] Z. Zhao and H. Liu, "Semi-supervised feature selection via spectral analysis," in *Proc. SIAM Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, 2007, pp. 641–646.

[235] M. Yang, Y.-J. Chen, and G.-L. Ji, "Semi_Fisher Score: A semi-supervised method for feature selection," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 1, Jul. 2010, pp. 527–532.

[236] H. Cheng, W. Deng, C. Fu, Y. Wang, and Z. Qin, "Graph-based semisupervised feature selection with application to automatic spam image identification," in *Proc. Int. Workshop Comput. Sci. Environ. Eng. Ecoinform.*, in Communications in Computer and Information Science, vol. 159. Berlin, Germany: Springer, 2011, pp. 259–264.

[237] J. Zhao, K. Lu, and X. He, "Locality sensitive semi-supervised feature selection," *Neurocomputing*, vol. 71, nos. 10–12, pp. 1842–1849, Jun. 2008.

[238] G. Doquire and M. Verleysen, "Graph Laplacian for semi-supervised feature selection in regression problems," in *Proc. Int. Work-Conf. Artif. Neural Netw.*, in Lecture Notes in Computer Science, vol. 6691. Berlin, Germany: Springer, 2011, pp. 248–255.

[239] G. Doquire and M. Verleysen, "A graph Laplacian based approach to semi-supervised feature selection for regression problems," *Neurocomputing*, vol. 121, pp. 5–13, Dec. 2013.

[240] M. Rostami, K. Berahmand, and S. Forouzandeh, "A novel method of constrained feature selection by the measurement of pairwise constraints uncertainty," *J. Big Data*, vol. 7, no. 1, pp. 1–21, 2020.

[241] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, and X. Zhou, "Semisupervised feature selection via spline regression for video semantic recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 252–264, Feb. 2015.

[242] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 1–9.

[243] Z. Zhao, L. Wang, and H. Liu, "Efficient spectral feature selection with minimum redundancy," in *Proc. AAAI Conf. Artif. Intell.*, 2010, vol. 24, no. 1, pp. 1–6.

[244] J. Ren, Z. Qiu, W. Fan, H. Cheng, and S. Y. Philip, "Forward semi-supervised feature selection," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, in Lecture Notes in Computer Science, vol. 5012. Berlin, Germany: Springer, 2008, pp. 970–976.

[245] H. Barkia, H. Elghazel, and A. Aussem, "Semi-supervised feature importance evaluation with ensemble learning," in *Proc. IEEE 11th Int. Conf. Data Mining*, Dec. 2011, pp. 31–40.

[246] F. Bellal, H. Elghazel, and A. Aussem, "A semi-supervised feature ranking method with ensemble learning," *Pattern Recognit. Lett.*, vol. 33, no. 10, pp. 1426–1433, 2012.

[247] Y. Han, K. Park, and Y.-K. Lee, "Confident wrapper-type semi-supervised feature selection using an ensemble classifier," in *Proc. 2nd Int. Conf. Artif. Intell., Manage. Sci. Electron. Commerce (AIMSEC)*, Aug. 2011, pp. 4581–4586.

[248] O. Chapelle, "'Semi-supervised learning, vol. 2,' Cambridge: MIT Press. Cortes, C., & Mohri, M.(2014). Domain adaptation and sample bias correction theory and algorithm for regression," *Theor. Comput. Sci.*, vol. 519, Jan. 2006, Art. no. 103126.

[249] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," 2014, *arXiv:1412.6596*.

[250] X. J. Zhu, "Semi-supervised learning literature survey," Univ. Wisconsin, Madison, WI, USA, 2005.

[251] V. Jothi Prakash and D. L. M. Nithya, "A survey on semi-supervised learning techniques," 2014, *arXiv:1402.4645*.

[252] Z. Ma, F. Nie, Y. Yang, J. R. R. Uijlings, N. Sebe, and A. G. Hauptmann, "Discriminating joint feature analysis for multimedia data understanding," *IEEE Trans. Multimedia*, vol. 14, no. 6, pp. 1662–1672, Dec. 2012.

[253] Z. Ma, Y. Yang, F. Nie, J. Uijlings, and N. Sebe, "Exploiting the entire feature space with sparsity for automatic image annotation," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 283–292.

[254] K. Benabdeslem and M. Hindawi, "Efficient semi-supervised feature selection: Constraint, relevance, and redundancy," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1131–1143, May 2014.

[255] R. Polikar, "Ensemble learning," in *Ensemble Machine Learning*. Berlin, Germany: Springer, 2012, pp. 1–34.

[256] A. A. Aburomman and M. B. I. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," *Comput. Secur.*, vol. 65, pp. 135–152, Mar. 2017.

[257] C. Ying, M. Qi-Guang, L. Jia-Chen, and G. Lin, "Advance and prospects of AdaBoost algorithm," *Acta Automatica Sinica*, vol. 39, no. 6, pp. 745–758, 2013.

[258] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers Neurorobot.*, vol. 7, p. 21, Dec. 2013.

[259] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3146–3154.

[260] M. Govindarajan, "Hybrid intrusion detection using ensemble of classification methods," *Int. J. Comput. Netw. Inf. Secur.*, vol. 6, no. 2, pp. 45–53, 2014.

[261] A. K. Shrivas and A. K. Dewangan, "An ensemble model for classification of attacks with feature selection based on KDD99 and NSL-KDD data set," *Int. J. Comput. Appl.*, vol. 99, no. 15, pp. 8–13, Aug. 2014.

[262] Z. Wang, H. Huang, and Y. Wang, "Fault diagnosis of planetary gearbox using multi-criteria feature selection and heterogeneous ensemble learning classification," *Measurement*, vol. 173, Mar. 2021, Art. no. 108654.

[263] S. Zhao, J. Meng, and Y. Luan, "LncRNA-encoded short peptides identification using feature subset recombination and ensemble learning," *Interdiscipl. Sci., Comput. Life Sci.*, vol. 14, pp. 101–112, Jul. 2021.

[264] R. Malhotra, A. Budhiraja, A. Kumar Singh, and I. Ghoshal, "A novel feature selection approach based on binary particle swarm optimization and ensemble learning for heterogeneous defect prediction," in *Proc. 3rd Asia Pacific Inf. Technol. Conf.*, 2021, pp. 115–121.

[265] H. Eom, Y. Son, and S. Choi, "Feature-selective ensemble learning-based long-term regional PV generation forecasting," *IEEE Access*, vol. 8, pp. 54620–54630, 2020.

[266] I. H. Witten, "Data mining: Practical machine learning tools and techniques with Java implementations," *Acm SIGMOD Rec.*, vol. 31, no. 1, pp. 76–77, 2016.

[267] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[268] C. Cortes and V. Vapnik, "Support vector machine," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[269] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, no. 2, pp. 131–163, Nov. 1997.

[270] A. Wang, N. An, G. Chen, L. Li, and G. Alterovitz, "Accelerating wrapper-based feature selection with K-nearest-neighbor," *Knowl.-Based Syst.*, vol. 83, pp. 81–91, Jul. 2015.

[271] W. M. Shaban, A. H. Rabie, A. I. Saleh, and M. Abo-Elsoud, "A new COVID-19 patients detection strategy (CPDS) based on hybrid feature selection and enhanced KNN classifier," *Knowl.-Based Syst.*, vol. 205, Oct. 2020, Art. no. 106270.

[272] M. Rogati and Y. Yang, "High-performing feature selection for text classification," in *Proc. 11th Int. Conf. Inf. Knowl. Manage.*, 2002, pp. 659–661.

[273] A. M. Bommert, "Integration of feature selection stability in model fitting," Ph.D. dissertation, Fac. Statist., TU Dortmund Univ., Dortmund, Germany, 2021.

[274] M. Toğaçar, B. Ergen, and Z. Cömert, "Waste classification using AutoEncoder network with integrated feature selection method in convolutional neural network models," *Measurement*, vol. 153, Mar. 2020, Art. no. 107459.

[275] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection using an improved Chi-square for Arabic text classification," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 2, pp. 225–231, Feb. 2020.

[276] E.-S. M. El-kenawy, A. Ibrahim, S. Mirjalili, M. M. Eid, and S. E. Hussein, "Novel feature selection and voting classifier algorithms for COVID-19 classification in CT images," *IEEE Access*, vol. 8, pp. 179317–179335, 2020.

[277] A. Joshuva, R. S. Kumar, S. Sivakumar, G. Deenadayalan, and R. Vishnuvardhan, "An insight on VMD for diagnosing wind turbine blade faults using C4.5 as feature selection and discriminating through multilayer perceptron," *Alexandria Eng. J.*, vol. 59, no. 5, pp. 3863–3879, 2020.

[278] A. A. Nagra, F. Han, Q. H. Ling, M. Abubaker, F. Ahmad, S. Mehta, and A. T. Apasiba, ''Hybrid self-inertia weight adaptive particle swarm optimisation with local search using C4.5 decision tree classifier for feature selection problems,'' *Connection Sci.*, vol. 32, no. 1, pp. 16–36, 2020.

[279] A. Lestari, ''Increasing accuracy of C4.5 algorithm using information gain ratio and adaboost for classification of chronic kidney disease,'' *J. Soft Comput. Explor.*, vol. 1, no. 1, pp. 32–38, 2020.

[280] R. S. Subramanian and D. Prabha, ''Customer behavior analysis using Naive Bayes with bagging homogeneous feature selection approach,'' *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 5, pp. 5105–5116, 2021.

[281] B. K. Khotimah, M. Miswanto, and H. Suprajitno, ''Optimization of feature selection using genetic algorithm in Naïve Bayes classification for incomplete data,'' *Int. J. Intell. Eng. Syst*, vol. 13, no. 1, pp. 334–343, 2020.

[282] M. G. Tadesse, M. Vannucci, and P. Liò, ''Identification of DNA regulatory motifs using Bayesian variable selection,'' *Bioinformatics*, vol. 20, no. 16, pp. 2553–2561, 2004.

[283] Y.-M. Cheung, ''$K*$-means: A new generalized k-means clustering algorithm,'' *Pattern Recognit. Lett.*, vol. 24, no. 15, pp. 2883–2893, 2003.

[284] A. P. Dempster, N. M. Laird, and D. B. Rubin, ''Maximum likelihood from incomplete data via the EM algorithm,'' *J. Roy. Stat. Soc., B (Methodol.)*, vol. 39, no. 1, pp. 1–22, 1977.

[285] D. H. Fisher, ''Knowledge acquisition via incremental conceptual clustering,'' *Mach. Learn.*, vol. 2, no. 2, pp. 139–172, 1987.

[286] C. Boutsidis, P. Drineas, and M. W. Mahoney, ''Unsupervised feature selection for the k-means clustering problem,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 153–161.

[287] A. Hashemi and M. B. Dowlatshahi, ''MLCR: A fast multi-label feature selection method based on K-means and L_2-norm,'' in *Proc. 25th Int. Comput. Conf., Comput. Soc. Iran (CSICC)*, Jan. 2020, pp. 1–7.

[288] H. Yu, G. Wen, J. Gan, W. Zheng, and C. Lei, ''Self-paced learning for K-means clustering algorithm,'' *Pattern Recognit. Lett.*, vol. 132, pp. 69–75, Apr. 2020.

[289] X.-H. Wang, Y. Zhang, X.-Y. Sun, Y.-L. Wang, and C.-H. Du, ''Multi-objective feature selection based on artificial bee colony: An acceleration approach with variable sample size,'' *Appl. Soft Comput.*, vol. 88, p. 106041, 2020.

[290] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, vol. 382. Hoboken, NJ, USA: Wiley, 2007.

[291] Q. Yang, X. Jia, X. Li, J. Feng, W. Li, and J. Lee, ''Evaluating feature selection and anomaly detection methods of hard drive failure prediction,'' *IEEE Trans. Rel.*, vol. 70, no. 2, pp. 749–760, Jun. 2021.

[292] V. A. Nguyen, H. A. Le Thi, and H. M. Le, ''A DCA based algorithm for feature selection in model-based clustering,'' in *Proc. Asian Conf. Intell. Inf. Database Syst.*, in Lecture Notes in Computer Science, vol. 12033. Cham, Switzerland: Springer, 2020, pp. 404–415.

[293] G. Du, J. Zhang, Z. Luo, F. Ma, L. Ma, and S. Li, ''Joint imbalanced classification and feature selection for hospital readmissions,'' *Knowl.-Based Syst.*, vol. 200, Jul. 2020, Art. no. 106020.

[294] X. Li, W. Chen, Q. Zhang, and L. Wu, ''Building auto-encoder intrusion detection system based on random forest feature selection,'' *Comput. Secur.*, vol. 95, Aug. 2020, Art. no. 101851.

[295] E. Vans, A. Patil, and A. Sharma, ''FEATS: Feature selection-based clustering of single-cell RNA-seq data,'' *Briefings Bioinf.*, vol. 22, no. 4, 2021, Art. no. bbaa306.

[296] D. Stanisavljevic, D. Cemernek, H. Gursch, G. Urak, and G. Lechner, ''Detection of interferences in an additive manufacturing process: An experimental study integrating methods of feature selection and machine learning,'' *Int. J. Prod. Res.*, vol. 58, no. 9, pp. 2862–2884, 2020.

[297] X. Zhu, J. Gan, G. Lu, J. Li, and S. Zhang, ''Spectral clustering via half-quadratic optimization,'' *World Wide Web*, vol. 23, pp. 1969–1988, Nov. 2020.

[298] G. Kou, P. Yang, Y. Peng, F. Xiao, Y. Chen, and F. E. Alsaadi, ''Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods,'' *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105836.

[299] H. Bostani and M. Sheikhan, ''Hybrid of binary gravitational search algorithm and mutual information for feature selection in intrusion detection systems,'' *Soft Comput.*, vol. 21, no. 9, pp. 2307–2324, May 2017.

[300] G. Liu, B. Zhang, X. Ma, and J. Wang, ''Network intrusion detection based on chaotic multi-verse optimizer,'' in *Proc. 11th EAI Int. Conf. Mobile Multimedia Commun.*, 2018, p. 218.

[301] F. Barani, M. Mirhosseini, and H. Nezamabadi-Pour, ''Application of binary quantum-inspired gravitational search algorithm in feature subset selection,'' *Appl. Intell.*, vol. 47, no. 2, pp. 304–318, 2017.

[302] A. A. Ewees, M. Aziz, and A. E. Hassanien, ''Chaotic multi-verse optimizer-based feature selection,'' *Neural Comput. Appl.*, vol. 31, no. 1, pp. 991–1006, 2019.

[303] A. K. Naik, V. Kuppili, and D. R. Edla, ''Efficient feature selection using one-pass generalized classifier neural network and binary bat algorithm with a novel fitness function,'' *Soft Comput.*, vol. 24, no. 6, pp. 4575–4587, 2020.

[304] M. Belazzoug, M. Touahria, F. Nouioua, and M. Brahimi, ''An improved sine cosine algorithm to select features for text categorization,'' *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 4, pp. 454–464, 2020.

[305] S. Nogueira and G. Brown, ''Measuring the stability of feature selection,'' in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, in Lecture Notes in Computer Science, vol. 9852. Cham, Switzerland: Springer, 2016, pp. 442–457.

[306] U. M. Khaire and R. Dhanalakshmi, ''Stability of feature selection algorithm: A review,'' *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1060–1073, 2019.

[307] M. H. Aghdam, N. Ghasem-Aghaee, and M. E. Basiri, ''Text feature selection using ant colony optimization,'' *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6843–6853, Apr. 2009.

[308] G. Forman, ''An extensive empirical study of feature selection metrics for text classification,'' *J. Mach. Learn. Res.*, vol. 3, pp. 1289–1305, Mar. 2003.

[309] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, and M. W. Mahoney, ''Feature selection methods for text classification,'' in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 230–239.

[310] G. Forman, ''Feature selection for text classification,'' in *Computational Methods of Feature Selection*. Boca Raton, FL, USA: CRC Press, 2007, pp. 273–292.

[311] K. Brkić, A. Pinz, S. Šegvić, and Z. Kalafatić, ''Histogram-based description of local space-time appearance,'' in *Proc. Scand. Conf. Image Anal.*, in Lecture Notes in Computer Science, vol. 6688. Berlin, Germany: Springer, 2011, pp. 206–217.

[312] P. P. Ohanian and R. C. Dubes, ''Performance evaluation for four classes of textural features,'' *Pattern Recognit.*, vol. 25, no. 8, pp. 819–833, Aug. 1992.

[313] A. Jain and D. Zongker, ''Feature selection: Evaluation, application, and small sample performance,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 153–158, Feb. 1997.

[314] B. Remeseiro, V. Bolon-Canedo, D. Peteiro-Barral, A. Alonso-Betanzos, B. Guijarro-Berdinas, A. Mosquera, M. G. Penedo, and N. Sánchez-Marono, ''A methodology for improving tear film lipid layer classification,'' *IEEE J. Biomed. Health Inform.*, vol. 18, no. 4, pp. 1485–1493, Jul. 2014.

[315] L. Wang and L. Khan, ''Automatic image annotation and retrieval using weighted feature selection,'' *Multimedia Tools Appl.*, vol. 29, no. 1, pp. 55–71, 2006.

[316] L. Setia and H. Burkhardt, ''Feature selection for automatic image annotation,'' in *Proc. Joint Pattern Recognit. Symp.*, in Lecture Notes in Computer Science, vol. 4174. Berlin, Germany: Springer, 2006, pp. 294–303.

[317] Y. Gao, J. Fan, X. Xue, and R. Jain, ''Automatic image annotation by incorporating feature hierarchy and boosting to scale up SVM classifiers,'' in *Proc. 14th ACM Int. Conf. Multimedia*, 2006, pp. 901–910.

[318] C. Jin and C. Yang, ''Integrating hierarchical feature selection and classifier training for multi-label image annotation,'' in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2011, pp. 515–524.

[319] J. Friedman, *The Elements of Statistical Learning* (Springer Series in Statistics), vol. 1, no. 10. New York, NY, USA: Springer, 2001.

[320] J. Lu, T. Zhao, and Y. Zhang, ''Feature selection based-on genetic algorithm for image annotation,'' *Knowl.-Based Syst.*, vol. 21, no. 8, pp. 887–891, 2008.

[321] J. Yang, D. Zhang, Y. Xu, and J. Yang, ''Two-dimensional discriminant transform for face recognition,'' *Pattern Recognit.*, vol. 38, no. 7, pp. 1125–1129, 2005.

[322] F. M. S. de Matos, L. V. Batista, and J. V. Poel, ''Face recognition using DCT coefficients selection,'' in *Proc. ACM Symp. Appl. Comput.*, 2008, pp. 1753–1757.

[323] S. H. Lee, J. Y. Choi, K. N. Plataniotis, and Y. M. Ro, ''Color component feature selection in feature-level fusion based color face recognition,'' in *Proc. Int. Conf. Fuzzy Syst.*, Jul. 2010, pp. 1–6.

[324] Z. Yan and C. Yuan, "Ant colony optimization for feature selection in face recognition," in *Proc. Int. Conf. Biometric Authentication*, in Lecture Notes in Computer Science, vol. 3072. Berlin, Germany: Springer, 2004, pp. 221–226.

[325] C. Qiu, "A novel multi-swarm particle swarm optimization for feature selection," *Genetic Program. Evolvable Mach.*, vol. 20, no. 4, pp. 503–529, 2019.

[326] R. M. Ramadan and R. F. Abdel-Kader, "Face recognition using particle swarm optimization-based selected features," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 2, no. 2, pp. 51–65, Jun. 2009.

[327] A. Amine, A. El Akadi, M. Rziza, and D. Aboutajdine, "GA-SVM and mutual information based frequency feature selection for face recognition," *INFOCOMP J. Comput. Sci.*, vol. 8, no. 1, pp. 20–29, 2009.

[328] L. D. Vignolo, D. H. Milone, and J. Scharcanski, "Feature selection for face recognition based on multi-objective evolutionary wrappers," *Expert Syst. Appl.*, vol. 40, no. 13, pp. 5077–5084, 2013.

[329] D. Mazumdar, S. Mitra, and S. Mitra, "Evolutionary-rough feature selection for face recognition," in *Transactions on Rough Sets XII* (Lecture Notes in Computer Science), vol. 619. Berlin, Germany: Springer, 2010, pp. 117–142.

[330] N. Gayatri, S. Nickolas, A. Reddy, S. Reddy, and A. Nickolas, "Feature selection using decision tree induction in class level metrics dataset for software defect predictions," in *Proc. World Congr. Eng. Comput. Sci.*, vol. 1, 2010, pp. 124–129.

[331] E. F. Petricoin and L. A. Liotta, "Mass spectrometry-based diagnostics: The upcoming revolution in disease detection," *Clin. Chem.*, vol. 49, no. 4, pp. 533–534, 2003.

[332] K.-J. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert Syst. Appl.*, vol. 19, no. 2, pp. 125–132, Aug. 2000.

[333] H. Liu and R. Setiono, "Dimensionality reduction via discretization," *Knowl.-Based Syst.*, vol. 9, no. 1, pp. 67–72, 1996.

[334] R. K. Bakshi, N. Kaur, R. Kaur, and G. Kaur, "Opinion mining and sentiment analysis," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2016, pp. 452–455.

[335] S. Doraisamy, S. Golzari, N. Mohd, M. N. Sulaiman, and N. I. Udzir, "A study on feature selection and classification techniques for automatic genre classification of traditional Malay music," in *Proc. ISMIR*, 2008, pp. 331–336.

[336] S. L. Salzberg, A. L. Delcher, S. Kasif, and O. White, "Microbial gene identification using interpolated Markov models," *Nucleic Acids Res.*, vol. 26, no. 2, pp. 544–548, 1998.

[337] A. L. Delcher, D. Harmon, S. Kasif, O. White, and S. L. Salzberg, "Improved microbial gene identification with GLIMMER," *Nucleic Acids Res.*, vol. 27, no. 23, pp. 4636–4641, 1999.

[338] S. Keleş, M. van der Laan, and M. B. Eisen, "Identification of regulatory elements using a feature selection method," *Bioinformatics*, vol. 18, no. 9, pp. 1167–1175, 2002.

[339] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, "Tissue classification with gene expression profiles," in *Proc. 4th Annu. Int. Conf. Comput. Mol. Biol.*, 2000, pp. 54–64.

[340] A. D. Baxevanis, G. D. Bader, and D. S. Wishart, *Bioinformatics*. Hoboken, NJ, USA: Wiley, 2020.

[341] J. Tang and H. Liu, "Feature selection with linked data in social media," in *Proc. SIAM Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, 2012, pp. 118–128.

[342] J. Tang and H. Liu, "Feature selection for social media data," *ACM Trans. Knowl. Discovery Data*, vol. 8, no. 4, pp. 1–27, 2014.

[343] J. Tang and H. Liu, "An unsupervised feature selection framework for social media data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 2914–2927, Dec. 2014.

[344] F. Wu, Y. Han, X. Liu, J. Shao, Y. Zhuang, and Z. Zhang, "The heterogeneous feature selection with structural sparsity for multimedia annotation and hashing: A survey," *Int. J. Multimedia Inf. Retr.*, vol. 1, no. 1, pp. 3–15, 2012.

[345] J. Ronald Eisenberg, R. L. Eisenberg, and A. Margulis, *A Patient's Guide to Medical Imaging*. Oxford, U.K.: Oxford Univ. Press, 2011.

[346] N. Linder, R. Turkki, M. Walliander, A. Mårtensson, V. Diwan, E. Rahtu, M. Pietikäinen, M. Lundin, and J. Lundin, "A malaria diagnostic tool based on computer vision screening and visualization of plasmodium falciparum candidate areas in digitized blood smears," *PLoS ONE*, vol. 9, no. 8, 2014, Art. no. e104855.

[347] M. J. Budoff and J. S. Shinbane, *Cardiac CT Imaging: Diagnosis of Cardiovascular Disease*. Springer, 2016.

[348] K. D. Fritscher, M. Peroni, P. Zaffino, M. F. Spadea, R. Schubert, and G. Sharp, "Automatic segmentation of head and neck CT images for radiotherapy treatment planning using multiple atlases, statistical appearance models, and geodesic active contours," *Med. Phys.*, vol. 41, no. 5, 2014, Art. no. 051910.

[349] A. K. Tiwari, R. B. Pachori, V. Kanhangad, and B. K. Panigrahi, "Automated diagnosis of epilepsy using key-point-based local binary pattern of EEG signals," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 4, pp. 888–896, Jul. 2017.

[350] I. Mporas, V. Tsirka, E. I. Zacharaki, M. Koutroumanidis, M. Richardson, and V. Megalooikonomou, "Seizure detection using EEG and ECG signals for computer-based monitoring, analysis and management of epileptic patients," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3227–3233, 2015.

[351] K. K. Ang, K. S. G. Chua, K. S. Phua, C. Wang, Z. Y. Chin, C. W. K. Kuah, W. Low, and C. Guan, "A randomized controlled trial of EEG-based motor imagery brain-computer interface robotic rehabilitation for stroke," *Clin. EEG Neurosci.*, vol. 46, no. 4, pp. 310–320, 2015.

[352] C. O. Sakar, O. Kursun, and F. Gurgen, "A feature selection method based on kernel canonical correlation analysis and the minimum redundancy–maximum relevance filter method," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3432–3437, 2012.

[353] M. Radovic, M. Ghalwash, N. Filipovic, and Z. Obradovic, "Minimum redundancy maximum relevance feature selection approach for temporal gene expression data," *BMC Bioinf.*, vol. 18, no. 1, pp. 1–14, 2017.

[354] C. Liu, D. Jiang, and W. Yang, "Global geometric similarity scheme for feature selection in fault diagnosis," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3585–3595, 2014.

[355] C. Liu, J. Yang, R. Chen, Y. Zhang, and J. Zeng, "Research on immunity-based intrusion detection technology for the Internet of Things," in *Proc. 17th Int. Conf. Natural Comput.*, vol. 1, Jul. 2011, pp. 212–216.

[356] W. Li, S. Tug, W. Meng, and Y. Wang, "Designing collaborative blockchained signature-based intrusion detection in IoT environments," *Future Gener. Comput. Syst.*, vol. 96, pp. 481–489, Jul. 2019.

[357] F. Erlacher and F. Dressler, "FIXIDS: A high-speed signature-based flow intrusion detection system," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2018, pp. 1–8.

[358] N. F. Haq, A. R. Onik, and F. M. Shah, "An ensemble framework of anomaly detection using hybridized feature selection approach (HFSA)," in *Proc. SAI Intell. Syst. Conf. (IntelliSys)*, Nov. 2015, pp. 989–995.

[359] M. Monshizadeh, V. Khatri, B. G. Atli, R. Kantola, and Z. Yan, "Performance evaluation of a combined anomaly detection platform," *IEEE Access*, vol. 7, pp. 100964–100978, 2019.

[360] A. I. Hajamydeen and N. I. Udzir, "A detailed description on unsupervised heterogeneous anomaly based intrusion detection framework," *Scalable Comput., Pract. Exper.*, vol. 20, no. 1, pp. 113–160, 2019.

[361] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou, "Specification-based anomaly detection: A new approach for detecting network intrusions," in *Proc. 9th ACM Conf. Comput. Commun. Secur.*, 2002, pp. 265–274.

[362] W. Yassin, N. I. Udzir, A. Abdullah, M. T. Abdullah, H. Zulzalil, and Z. Muda, "Signature-based Anomaly intrusion detection using integrated data mining classifiers," in *Proc. Int. Symp. Biometrics Secur. Technol. (ISBAST)*, Aug. 2014, pp. 232–237.

[363] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Comput. Netw.*, vol. 136, pp. 37–50, May 2018.

[364] Y. Zhai, Y.-S. Ong, and I. W. Tsang, "The emerging 'big dimensionality,'" *IEEE Comput. Intell. Mag.*, vol. 9, no. 3, pp. 14–26, Aug. 2014.

[365] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling up Machine Learning: Parallel and Distributed Approaches*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[366] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "Recent advances and emerging challenges of feature selection in the context of big data," *Knowl.-Based Syst.*, vol. 86, pp. 33–45, Sep. 2015.

[367] E. Alba, J. Garcia-Nieto, L. Jourdan, and E.-G. Talbi, "Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 284–290.

[368] F. Tan, X. Fu, Y. Zhang, and A. G. Bourgeois, "A genetic algorithm-based method for feature subset selection," *Soft Comput.*, vol. 12, no. 2, pp. 111–120, 2008.

[369] S. Ahmed, M. Zhang, and L. Peng, "Improving feature ranking for biomarker discovery in proteomics mass spectrometry data using genetic programming," *Connection Sci.*, vol. 26, no. 3, pp. 215–243, 2014.

[370] L. Y. Chuang, H. W. Chang, C. J. Tu, and C. H. Yang, "Improved binary PSO for feature selection using gene expression data," *Comput. Biol. Chem.*, vol. 32, pp. 29–38, Feb. 2008.

[371] B. Sahu and D. Mishra, "A novel feature selection algorithm using particle swarm optimization for cancer microarray data," *Proc. Eng.*, vol. 38, no. 5, pp. 27–31, 2012.

[372] S. I. Ali and W. Shahzad, "A feature subset selection method based on symmetric uncertainty and ant colony optimization," in *Proc. Int. Conf. Emerg. Technol.*, Oct. 2012, pp. 1–6.

[373] M. Tan, I. W. Tsang, and L. Wang, "Towards ultrahigh dimensional feature selection for big data," *J. Mach. Learn. Res.*, vol. 15, pp. 1371–1429, Jan. 2014.

[374] S. Alelyani, Z. Zhao, and H. Liu, "A dilemma in assessing stability of feature selection algorithms," in *Proc. IEEE Int. Conf. High Perform. Comput. Commun.*, Sep. 2011, pp. 701–707.

[375] J. Li, J. Tang, and H. Liu, "Reconstruction-based unsupervised feature selection: An embedded approach," in *Proc. IJCAI*, 2017, pp. 2159–2165.

[376] J. Li and H. Liu, "Challenges of feature selection for big data analytics," *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 9–15, 2017.

[377] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Appl. Soft Comput.*, vol. 18, pp. 261–276, May 2014.

[378] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, May 2003.

[379] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "An ensemble of filters and classifiers for microarray data classification," *Pattern Recognit.*, vol. 45, no. 1, pp. 531–539, 2012.

[380] V. Bolón-Canedo, N. Sánchez-Marono, and A. Alonso-Betanzos, "Data classification using an ensemble of filters," *Neurocomputing*, vol. 135, pp. 13–20, Jul. 2014.

[381] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[382] Wikipedia Contributors. (2021). *Apache Hadoop—Wikipedia, the Free Encyclopedia*. Accessed: Sep. 6, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Apache_Hadoop&oldid=1039287079

[383] (2021). *Apache Spark—Wikipedia, the Free Encyclopedia*. Accessed: Sep. 6, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Apache_Spark&oldid=1040611519

[384] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar, "MLlib: Machine learning in apache spark," Tech. Rep., 2015.

[385] O. Fontenla-Romero, B. Guijarro-Berdiñas, D. Martinez-Rego, B. Pérez-Sánchez, and D. Peteiro-Barral, "Online machine learning," in *Efficiency and Scalability Methods for Computational Intellect*. Hershey, PA, USA: IGI Global, 2013, pp. 27–54.

[386] C. Zhang, J. Ruan, and Y. Tan, "An incremental feature subset selection algorithm based on Boolean matrix in decision system," *Converg. Inf. Technol.*, pp. 16–23, 2011.

[387] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.

[388] T. Butler-Yeoman, B. Xue, and M. Zhang, "Particle swarm optimisation for feature selection: A hybrid filter-wrapper approach," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 2428–2435.

[389] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Comput. Appl.*, vol. 24, no. 1, pp. 175–186, Jan. 2014.

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

[391] L. Cervante, B. Xue, M. Zhang, and L. Shang, "Binary particle swarm optimisation for feature selection: A filter based approach," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8.

[392] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[393] Y. Zhai, M. Tan, I. Tsang, and Y. Soon Ong, "Discovering support and affiliated features from very high dimensions," 2012, *arXiv:1206.6477*.

[394] X. Wang, J. Yang, and X. Teng, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognit. Lett.*, vol. 28, no. 4, pp. 459–471, 2007.

[395] B. Xue, L. Cervante, L. Shang, W. N. Browne, and M. Zhang, "Binary PSO and rough set theory for feature selection: A multi-objective filter based approach," *Int. J. Comput. Intell. Appl.*, vol. 13, no. 2, 2014, Art. no. 1450009.

[396] Q. Mao and I. W.-H. Tsang, "A feature selection method for multivariate performance measures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2051–2063, Sep. 2013.

[397] S. Wang, W. Pedrycz, Q. Zhu, and W. Zhu, "Subspace learning for unsupervised feature selection via matrix factorization," *Pattern Recognit.*, vol. 48, no. 1, pp. 10–19, 2015.

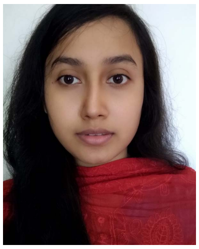
[398] K. Bunte, M. Biehl, and B. Hammer, "A general framework for dimensionality-reducing data visualization mapping," *Neural Comput.*, vol. 24, no. 3, pp. 771–804, Dec. 2011.

[399] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[400] P. Flach, *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[401] J. Krause, A. Perer, and E. Bertini, "INFUSE: Interactive feature selection for predictive modeling of high dimensional data," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1614–1623, Dec. 2014.

MD RASHEDUL ISLAM (Member, IEEE) received the B.Sc. degree in computer science and engineering from the University Rajshahi, Rajshahi, Bangladesh, in 2006, the M.Sc. degree in informatics from the Högskolan i Borås (University of Boras), Boras, Sweden, in 2011, and the Ph.D. degree in electrical, electronic, and computer engineering from the University of Ulsan, Ulsan, South Korea, in 2016. He is currently working as a Senior Architect with the Research and Development Department, Exvision Corporation, Tokyo, Japan, and also an Associate Professor (on leave) with the Department of Computer Science and Engineering, University of Asia Pacific (UAP), Dhaka, Bangladesh. Previously, he worked as a Visiting Researcher (Postdoctoral Researcher) with the School of Computer Science and Engineering, The University of Aizu, Japan; a Graduate Research Assistant with the Embedded System Laboratory, University of Ulsan, South Korea; an Assistant Professor with the Department of Computer Science and Engineering, University of Asia Pacific (UAP), Dhaka, Bangladesh; and a Lecturer with the Department of Computer Science and Engineering, Leading University, Sylhet, Bangladesh. His research interests include machine learning, signal & image processing, HCI, health informatics, bearing fault diagnosis, and others. He is also a PC member of several international conferences. Also, he has a good experience in professional IT system analysis and development. He is a member of the IEEE Computer Society and the IEEE Computational Intelligence Society. He has also served as the Secretary of the Organizing Committee of the 19th International Conference on Computer and Information Technology 2017 (ICCIT2017), an Organizing Chair of the Organizing Committee of the ACM-ICPC Dhaka Regional Site 2017, the Head of the Self-Assessment Committee (SAC) of the Department of CSE under IQAC, University of Asia Pacific, a Co-ordinator of the MCSE Program, Department of CSE, University of Asia Pacific, a Convener of Software and Hardware Club, Department of CSE, University of Asia Pacific, a Co-ordinator of the Admission Committee, Department of CSE, University of Asia Pacific, and a Treasurer of the Bangladesh Advanced Computing Society. He is a Reviewer of several journals, such as the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE ACCESS, *Applied Science*, *Multimedia Tools and Applications*, *Cluster Computing*, *Shock and Vibration*, *Journal of Information Processing Systems*, and others.



AKLIMA AKTER LIMA is currently pursuing the degree in computer science with the Bangladesh University of Business and Technology. She is well organized, enthusiastic, and determined to work. She worked as a Research Assistant with the Advanced Machine Learning Laboratory. She is also working as a Research Engineer with the Computing & Advanced Intelligence Laboratory. She has experience working with Tensorflow, Keras, Matplotlib, and Numpy. Her research experience includes advanced driver assistance systems, stock exchange, automatic text summarization, brain–computer interface, and unsupervised writer identification. Her research interests include machine learning, image preprocessing, deep learning, natural language processing (NLP).

SUJOY CHANDRA DAS is currently pursuing the degree in computer science and engineering with the Bangladesh University of Business and Technology. He worked as an Assistant Researcher with the Advanced Machine Learning Laboratory. He is determinant, communicative, and sincere to work. He has good communication and presentation skills too. He has experience working with front-end web development, Tensorflow, Keras, and Matplotlib. He is interested in deep learning research. His research interests include machine learning, HCI, and NLP. He is also working with research topics, such as brain–computer interface, advanced driver assistance systems, automatic text summarization, feature selection, and writer identification.

M. F. MRIDHA (Senior Member, IEEE) received the Ph.D. degree in AI/ML from Jahangirnagar University, in 2017. He joined the Department of Computer Science and Engineering, Stanford University, Bangladesh, as a Lecturer, in June 2007. He was promoted as a Senior Lecturer at the Department of Computer Science and Engineering, Stanford University, in October 2010, and promoted as an Assistant Professor at the Department of Computer Science and Engineering, Stanford University, in October 2011. Then, he joined UAP, as an Assistant Professor, in May 2012. He also worked as a Faculty Member at the CSE Department, University of Asia Pacific, and as a Graduate Co-ordinator, from 2012 to 2019. He is currently working as an Associate Professor with the Department of Computer Science and Engineering, Bangladesh University of Business and Technology. His research experience, within both academia and industry, results in over 80 journals and conference publications. For more than ten years, he has been with the master's and undergraduate students as a supervisor of their thesis work. His research interests include artificial intelligence (AI), machine learning, natural language processing (NLP), and big data analysis. He has served as a program committee member in several international conferences/workshops. He served as an associate editor of several journals.

AKIBUR RAHMAN PRODEEP is currently pursuing the degree in computer science and engineering with the Bangladesh University of Business and Technology. He is also working as a Researcher Assistant with the Advanced Machine Learning Laboratory. He is an optimistic, energetic, enthusiastic, and devoted individual searching out a challenging situation to effectively use his assembled knowledge of artificial intelligence. He has experience working with Python, Tensorflow, Keras, Matplotlib, Numpy, and Pandas. His research interests include deep learning, image processing, computer vision, and natural language processing. He is also working on lung nodule and cancer recognition, feature selection, acne and rosacea (skin diseases) detection, and plant diseases identification.

YUTAKA WATANOBE (Member, IEEE) received the master's and Ph.D. degrees from The University of Aizu, Japan, in 2004 and 2007, respectively. He was a Research Fellow of the Japan Society for the Promotion of Science (JSPS), The University of Aizu, in 2007. He is currently a Senior Associate Professor at The University of Aizu. His research interests include visual programming, smart learning, data mining, and robotics.