A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects

Article *in* IEEE Access · September 2022

DOI: 10.1109/ACCESS.2022.3207287

2 authors:

Domor I Mienye
University of Johannesburg
25 PUBLICATIONS 1,085 CITATIONS
SEE PROFILE

Yanxia Sun
University of Johannesburg
183 PUBLICATIONS 3,682 CITATIONS
SEE PROFILE

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 August 2022, accepted 13 September 2022, date of publication 16 September 2022,
date of current version 26 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3207287

SURVEY

A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects

IBOMOIYE DOMOR MIENYE, (Member, IEEE), AND YANXIA SUN, (Senior Member, IEEE)
Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg 2006, South Africa

Corresponding author: Yanxia Sun (ysun@uj.ac.za)

This work was supported in part by the South African National Research Foundation under Grant 120106 and Grant 132797, and in part by the South African National Research Foundation Incentive under Grant 132159.

ABSTRACT Ensemble learning techniques have achieved state-of-the-art performance in diverse machine learning applications by combining the predictions from two or more base models. This paper presents a concise overview of ensemble learning, covering the three main ensemble methods: bagging, boosting, and stacking, their early development to the recent state-of-the-art algorithms. The study focuses on the widely used ensemble algorithms, including random forest, adaptive boosting (AdaBoost), gradient boosting, extreme gradient boosting (XGBoost), light gradient boosting machine (LightGBM), and categorical boosting (CatBoost). An attempt is made to concisely cover their mathematical and algorithmic representations, which is lacking in the existing literature and would be beneficial to machine learning researchers and practitioners.

INDEX TERMS Algorithms, classification, ensemble learning, fraud detection, machine learning, medical diagnosis.

I. INTRODUCTION

The recent advances in computing power and machine learning (ML) have given rise to several innovations and developments in many areas of research and human lives. In the last few years, advancements in machine learning, a subset of artificial intelligence (AI), have transformed and inherently changed almost every area of our lives [1], [2]. Particularly, ML has been applied in disease diagnosis [3], fraud detection [4], text classification [5], and image recognition [6], among others. Unlike humans, ML algorithms consider several factors when making decisions, and they are not prone to fatigue or prejudice. Meanwhile, learning can sometimes be challenging, especially when learning from high-dimensional and imbalanced datasets [7],[8], [9]. Research has shown that conventional ML algorithms tend to underperform when trained with imbalanced datasets [10]. Therefore, researchers have frequently resorted to new and improved learning approaches, such as ensemble and deep learning.

Ensemble learning and deep learning are two approaches that have dominated the machine learning domain [11], [12], [13]. Ensemble learning methods train multiple base learners and combine their predictions to obtain improved performance and better generalization ability than the individual base learners [14]. The fundamental idea behind ensemble learning is the recognition that machine learning models have limitations and can make errors. Hence, ensemble learning aims to improve classification performance by harnessing the strengths of multiple base models. Meanwhile, some limitations of ML algorithms include: that they result in models with high variance, high bias, and low accuracy [15], [16]. However, several studies have shown that ensemble models often achieve higher accuracy than single ML models [17]. Ensemble methods can limit the variance and bias errors associated with single ML models; for example, bagging reduces variance without increasing the bias, while boosting reduces bias [18], [19], [20]. Overall, ensemble classifiers are more robust and perform better than the individual ensemble learners.

Ensemble learning methods are broadly categorized into boosting, bagging, and stacking [21]. Gradient boosting, XGBoost, and AdaBoost are examples of boosting

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

algorithms, while random forest and Extra Trees classifier are well-known bagging algorithms. Meanwhile, examples of the stacking framework include super ensemble and blending techniques [22], [23], [24]. These ensemble algorithms have achieved excellent performance in various ML applications [25], [26], [27]. Their utilization in many real-world applications is also well-known [28]. Due to their high performance, ensemble methods are the go-to algorithms in many machine learning competitions. When implementing ensemble classifiers, the accuracy and diversity of the base learners are two essential factors that need to be considered [29]. Most ensemble algorithms ensure diversity through data resampling or by altering the structure of the individual learners [30]. Furthermore, base learners that ensure higher accuracy than random guessing are desirable. The individual learners need to have separate knowledge of the task being learned and are also expected to have errors different from the others.

In the past, researchers have conducted some general ensemble learning reviews [31], [32], [33], reviews that focus on ensemble learning for feature selection [34], reviews on ensemble learning in healthcare applications [35], and a review of recent developments and applications of ensemble learning [36]. Other ensemble learning reviews in the literature include those that focus on intrusion detection systems [37], sentiment analysis [38], and big data [39]. A taxonomy for characterizing ensemble methods was presented in [40]. Most published reviews and surveys provide a general overview of ensemble learning and focus on its applications in specific tasks. However, what is lacking in the literature is a succinct explanation of ensemble learning algorithms, their early developments, and concise mathematical and algorithmic representations in one peer-reviewed article. We think it is relevant and timely to have a survey that fills this gap, considering the rise in the popularity of ensemble methods, their application in diverse problems, and the introduction of more recent variants such as XGBoost, LighGBM, and CatBoost.

Therefore, this paper provides a concise yet deep insight into ensemble learning methods and algorithms. The paper aims to differentiate between the three main ensemble methods: bagging, boosting, and stacking. Particular focus is placed on the mathematical and algorithmic representations of the widely used ensemble algorithms, including random forest, gradient boosting, AdaBoost, XGBoost, LightGBM, and CatBoost. This review is for ML researchers and practitioners who intend to understand ensemble learning and the various ensemble algorithms.

The remainder of this paper comprises the following sections: Section 2 introduces the concept of ensemble learning, its early development, the main building blocks of an ensemble classifier, the popular methods used to combine ensemble base learners, and ensemble selection strategies. Meanwhile, Section 3 discusses the various ensemble methods and algorithms. Section 4 outlines recent ensemble learning applications, focusing on medical diagnosis, fraud detection, and sentiment analysis. Meanwhile, Section 5 presents a discussion and future research directions, while Section 6 concludes the paper.

II. OVERVIEW OF ENSEMBLE LEARNING

This section presents an overview of ensemble learning, detailing the building blocks of most ensemble methods, the techniques used for combining ensemble base learners, and ensemble selection methods. Most research works refer to the 1979 article by Dasarathy and Sheela [41] as one of the foundation works of ensemble learning. The authors presented a method to partition the feature space using multiple component classifiers. Then in 1990, Hansen and Salamon [42] demonstrated that applying an ensemble of similar artificial neural network classifiers achieved superior prediction performance than a single classifier. Meanwhile, in the same year, Schapire [43] proposed the boosting technique, a method developed to convert a weak classifier into a strong one, and this technique was the foundation that gave rise to the present robust algorithms such as AdaBoost, gradient boosting, and XGBoost [33].

Ensemble learning is a technique used to combine two or more ML algorithms to obtain superior performance compared to when the constituent algorithms are used individually. Instead of relying on a single model, the predictions from the individual learners are combined using a combination rule to obtain a single prediction that is more accurate. Generally, ensemble methods can be classified into parallel and sequential ensembles. The parallel methods train different base classifiers independently and combine their predictions using a combiner. A popular parallel ensemble method is bagging and its extension, the random forest algorithm [44]. Parallel ensemble algorithms use the parallel generation of base learners to encourage diversity in the ensemble members.

Meanwhile, sequential ensembles do not fit the base models independently. They are trained iteratively so that the models at every iteration learn to correct the errors made by the previous model. A popular type of sequential ensemble is the boosting algorithm [45]. Figures 1 and 2 show block diagrams depicting parallel and sequential ensemble learning. Furthermore, parallel ensembles can be classified into homogeneous or heterogeneous, depending on the base learners' homogeneity. Homogeneous ensembles consist of models built using the same ML algorithm, while heterogeneous ensembles comprise models from different algorithms [46], [47], [48].

The success of ensemble learning techniques mainly relies on the accuracy and diversity of the base learners [49]. A machine learning model is considered accurate if it has a good generalization ability on unseen instances. In contrast, ML models are diverse if their errors on unseen instances are not the same [47]. Therefore, diversity is seen as the difference between base learners in an ensemble [50]. Unlike accuracy, there is no general rule of thumb in measuring diversity. Meanwhile, it is challenging to have diversity in the base models when implementing ensemble classifiers. In most

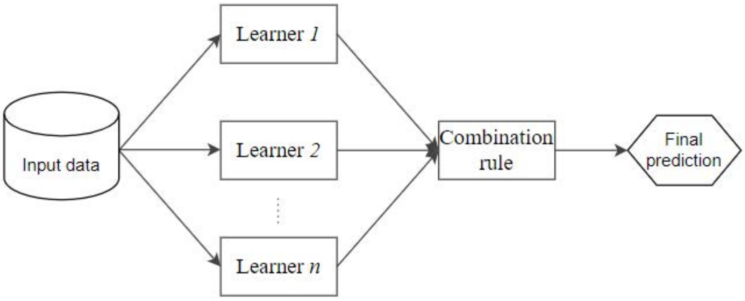


FIGURE 1. Block diagram of parallel ensemble learning.

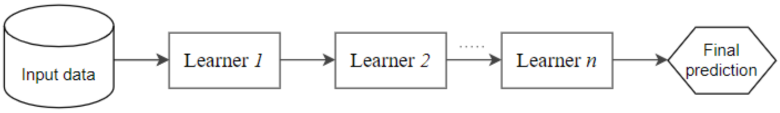


FIGURE 2. Block diagram of sequential ensemble learning.

ensembles, the base learners are trained using subsets of the same training data, making the models correlated and difficult to achieve diversity. Different ensemble techniques try to achieve diversity heuristically or implicitly. For instance, bagging achieves diversity by subsampling the training data while boosting achieves diversity by reweighting the training data.

Furthermore, different techniques are used to achieve diversity among base learners in homogeneous and heterogeneous ensembles. For example, heterogeneous ensembles employ different ML algorithms as base learners; therefore, they are essentially diverse. The main challenge in heterogeneous ensembles is obtaining the most effective method to combine the different base learners' predictions [46]. However, the main challenge of homogeneous ensemble methods is ensuring the base learners are diverse even though they use the same ML algorithm. Hence, bootstrap methods such as random forest [51] and boosting methods such as AdaBoost [52] have been developed to achieve diversity in the ensemble.

A. COMBINING BASE LEARNERS

A vital step in building ensemble classifiers is the method applied to combine the constituent base learners. The type of ensemble learning method usually determines the combination mechanism used. For example, the combination rule can be applied in bagging and boosting once the base models are trained, while stacking involves training a separate algorithm to do the combination [53]. The most common mechanism used to combine ensemble base models is majority voting.

1) MAJORITY VOTING

Majority voting is the most popular and intuitive combination method in classification and regression tasks [54]. In classification problems, the predictions for each class are summed, and the class with the majority vote is returned as the ensemble prediction. Meanwhile, the majority vote is achieved in regression tasks by computing the average predictions from the various base learners [33]. Assuming the decision of the $t-th$ classifier is $d_{t,c} \in \{0,1\}$, $t = 1, \ldots, T$, and $c = 1, \ldots, C$, where T and C represents the number of classifiers and the number of classes, respectively. Then, using majority voting, class ω_c* is selected as the ensemble prediction, if

$$\sum_{t=1}^{T} d_{t,c} = \max_{c} \sum_{t=1}^{T} d_{t,c} \tag{1}$$

Meanwhile, the ensemble's individual base models often do not have equal performance. Hence, considering them equally in the summation might be inappropriate [55]. A more suitable solution is to weigh the performance of the individual models using the weighted majority voting technique.

2) WEIGHTED MAJORITY VOTING

The weighted majority voting assumes that some classifiers in the ensemble are more skilful than others, and their predictions are given more priority when computing the final ensemble prediction. The conventional majority voting assumes that all the base models are equally skilled and their predictions are treated equally when calculating the final ensemble prediction. However, the weighted majority voting assigns a specific weight to the base classifiers, which is then multiplied by the models' output when computing the final ensemble prediction [56]. Assuming the generalization ability of each base model is known, then a weight W_t can be assigned to classifier h_t according to its estimated generalization ability. Therefore, using the weighted majority voting, the ensemble classifier selects class c^*, if

$$\sum_{t=1}^{T} w_t d_{t,c^*} = \max_{c} \sum_{t=1}^{T} w_t d_{t,c} \tag{2}$$

Usually, the voting weights are normalized so that their sum is equal to 1. A more comprehensive overview of the weighted majority voting is presented in [57]. Meanwhile, there are several combination rules in the literature. In [58], some combination rules were introduced, including the minimum, maximum, product, median, and sum rules.

B. ENSEMBLE SELECTION

Ensemble selection is a technique used to build ensemble classifiers from a set of base models. It is a vital topic in ensemble learning because selecting a suitable subset of base models could lead to better performance than when all the models are used to construct the ensemble classifier [59]. Since the base models are developed using various ML algorithms or different subsets of the training data, their performances would be different; while some would have good performance, others might have poor performance. Instead of combining the good and bad models, selecting only a subset of models with good performance might be beneficial, which would enhance the overall ensemble performance [60]. An ensemble selection strategy is employed to select the optimal subset of base classifiers, and they are usually guided by a scoring function [61]. Caruana *et al.* [60] developed the foremost forward model selection strategy to extract the best performing subset of base models, and its basic procedure is outlined as follows:

i. Begin with the empty ensemble.
ii. Select the base classifier from the library that maximizes the ensemble's performance using a validation set.
iii. Repeat step II for a predefined number of iterations or until all the models in the library have been examined.
iv. Return the subset of models that produces the best performance on the validation set.

The forward model selection strategy is fast and efficient but occasionally overfits, leading to poor ensemble performance. Hence, several ensemble selection strategies have been recently proposed; for example, Sun and Pfahringer [61] developed the bagging ensemble selection, which combines bagging and ensemble selection. Furthermore, the ensemble selection strategies can be divided into static and dynamic methods. The static strategy selects a single subset of base models during model training and applies it to predict all the unseen instances. The static selection methods can be grouped into ordering-based techniques and optimization-based techniques.

As the name implies, ordering-based techniques attempt to order the base models with respect to specific criteria, and only the top models are chosen as the optimal subset. Some criteria for ordering the base models include validation error and kappa measure [59]. Guo *et al.* [62] recently developed a method to order the base models via an evaluation metric that takes the margin and diversity into consideration. Meanwhile, optimization-based techniques formulate the selection process as an optimization problem that can be solved using mathematical programming or heuristic optimization. Static methods limits the flexibility of the ensemble selection process.

The dynamic methods dynamically select a subset of models for making a prediction based on specific features of the unseen instances. Every region of the input feature space is assigned a subset of models that perform best in that region. A foremost example of the dynamic selection strategy is the k-Nearest Neighbor Oracle (KNORA) [63]. Recently, Nguyen *et al.* [59] proposed a selection strategy that combines aspects of both static and dynamic ensemble selection. Other recently proposed dynamic ensemble selection methods can be found in [64], [65], [66]. Meanwhile, Pérez-Gállego *et al.* [67] present a detailed description of ensemble selection strategies and a comparison between static and dynamic selection methods.

III. ENSEMBLE LEARNING METHODS

A. BOOSTING

Boosting is a machine learning technique capable of converting weak learners into a strong classifier. It is a type of ensemble meta-algorithm used for reducing bias and variance. Meanwhile, a weak learner is a classifier that performs a bit better than random guessing, while strong learners are those that attain good accuracy [16], and they are the core of which the boosting ensemble algorithms are built. The boosting algorithm was first discussed in a 1990 paper by Schapire [43] in response to a question asked by Kearns and Valiant [68] if a set of weak learners could produce a single strong learner. The work by Schapire [43] had a significant impact on machine learning and statistics, which led to the development of several boosting algorithms, including AdaBoost [69] and XGBoost [70].

The main idea behind boosting involves iteratively applying the base learning algorithm to adjusted versions of the input data [14]. In particular, boosting techniques use the input data to train a weak learner, compute the predictions from the weak learner, select misclassified training samples, and train the subsequent weak learner with an adjusted training set comprising the misclassified instances from the previous training round [71]. The iterative learning process continues until a predefined number of base learners is obtained, and the base learners are weighted together [27]. Boosting focuses more on reducing bias than variance [28]. Therefore, it enhances base learners with a high bias and low variance, such as decision stumps (a decision tree with one internal node). Misclassified samples get more weight, causing the base learner to focus on such samples. Hence, if the base classifier is biased against specific samples, those samples are given more weight; hence, the algorithm corrects the bias. However, this iterative learning approach makes boosting unsuitable for learning noisy data because the weight given to noisy samples is usually much greater than the weights given to the other samples, thereby forcing the algorithm to focus excessively on the noisy samples, resulting in overfitting. Despite that, boosting-based ensemble methods are among the most successful algorithms in applied machine learning [72], [73].

1) AdaBoost

The AdaBoost algorithm is a type of boosting algorithm capable of using weak learners to obtain a robust classifier. It was developed in 1995 by Freund and Schapire [69] and is among the most robust ML algorithms. AdaBoost was the first successful boosting algorithm, and the base learners are decision trees having a single split. Because the decision trees are short, they are usually called decision stumps. The most successful AdaBoost implementation is the AdaBoost M1, used for binary classification tasks [74].

The AdaBoost learning process involves training a base classifier using a base algorithm, usually a decision tree. The sample weights are adjusted with respect to the classifier's predictions, and the adjusted samples are employed for training the subsequent classifier. Therefore, the misclassified samples are assigned larger weights and correctly classified instances are assigned lesser weights, ensuring that subsequent classifiers give more attention to the misclassified samples.

The different base learners are added sequentially and weighed to obtain the strong classifier [75]. At every iteration, the AdaBoost algorithm assigns weights to each instance in the training set [76]. Given m labelled training instances $S = \{(x_1, y_1), \dots, (x_i, y_i), \dots, (x_m, y_m)\}$, where y_i is the target label of sample x_i, and $y_i \in Y = \{-1, +1\}$, the weight D_1 of the sample x_i and the weight update D_{t+1} are computed as:

$$D_1(i) = \frac{1}{n}, i = 1, 2, \dots, m \tag{3}$$

$$D_{t+1}(i) = \frac{D_t(i)}{z_t} \exp(-\alpha_t y_i h_t(x_i)), \quad i = 1, 2, \dots, \tag{4}$$

where $h_t(x)$ is the base classifier, $t = 1, \dots, T$ is the number of iterations, Z_t denotes a normalization factor, while α_t is the weight of the classifier $h_t(x)$. The weight α_t measures the importance of the classifier $h_t(x)$ when obtaining the final classifier prediction. The instances that are wrongly predicted in $h_t(x)$ are assigned larger weights in the $t + 1$ training round. Furthermore, Z_t is selected such that D_{t+1} will be a distribution. Z_t and α_t are obtained using:

$$Z_t = \sum_{t+1}^{n} D_t(i) \exp\left(-\alpha_t y_i h_t(x_i)\right) \tag{5}$$

$$\alpha_t = \frac{1}{2} In(\frac{1 - \epsilon_t}{\epsilon_t}) \tag{6}$$

where ϵ_t represents the error rate of the classifier, and it is obtained using:

$$\epsilon_t = P[h_t(x_i) \neq y_i] = \sum_{i=n}^{n} D_i(i) I[h_t(x_i) \neq y_i] \tag{7}$$

When the given number of iterations have been completed, the final strong classifier is computed using:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)) \tag{8}$$

The AdaBoost algorithm is summarized in Algorithm 1. The AdaBoost is easy to implement with little need to tune its hyperparameters [77]. Furthermore, the AdaBoost is flexible and can use a variety of algorithms as the base learner; hence, an algorithm suitable for a specific application can be used as the base learner, and the AdaBoost can enhance its performance. However, a limitation of the AdaBoost is that it is sensitive to noisy data and outliers because of its iterative learning approach, causing overfitting.

```
Algorithm 1 AdaBoost.M1 Algorithm
Input: training data $S = (x_1, y_1), \dots, (x_2, y_2), \dots, (x_m, y_m)$
    The base algorithm $L$
    The number of iterations $T$.
    Procedure:
    Initialize the weight $D_1$ of sample $x_i$ using (3)
    for $t = 1, \dots, T$ :
      1) Train the base classifier $h_t(x)$ by minimizing $\epsilon_t$.
      2) Calculate the weight $\alpha_t$ of the classifier using (6).
      3) Update the sample weights using (4)
    end for
Output: Apply (8) to combine the predictions of the base
    classifiers to obtain the final strong classifier $H(x)$.
```

2) GRADIENT BOOSTING

Gradient boosting is a machine learning algorithm that uses the boosting technique to create strong ensembles. It mainly uses decision trees as the base learner to produce a robust ensemble classifier, and it is also called gradient boosted decision tree (GBDT). The gradient boosting technique was first introduced by Breiman [19], who noted that boosting can be represented as an optimization technique on an appropriate loss function.

Subsequently, an extended version of the gradient boosting algorithm was developed by Friedman [78]. The learning process of this algorithm involves sequentially training new models to obtain a robust classifier. It is developed in a step-by-step manner similar to other boosting techniques, but its core idea is to develop base learners that are highly correlated with the negative gradient of the loss function related to the entire ensemble [79].

Given a training set $S = \{x_i, y_i\}_1^N$, the gradient boosting technique aims to find an approximation, $\hat{F}(x)$, of a function $F^*(x)$ that maps the predictor variables x to their response variables y, by minimizing the specified loss function $L(y, F(x))$. The GBDT creates an additive approximation of $F^{(}x)$ through a weighted sum of functions:

$$F_m(x) = F_{m-1}(x) + \rho_m h_m(x) \tag{9}$$

where ρ_m represents the weight of the m^{th} function, $h_{m(x)}$. These functions are the decision tree models in the ensemble. The algorithm performs the approximation iteratively. Meanwhile, a constant approximation of $F^*(x)$ is achieved using:

$$F_0(x) \ = argmin_\alpha \sum_{i=1}^{N} L(y_i, \alpha) \tag{10}$$

Successive base learners aim to minimize

$$(\rho_m h_m(x)) = argmin_{\rho,h} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \rho h(x_i)) \tag{11}$$

Meanwhile, rather than performing the optimization task directly, every h_m can be considered a greedy step in a gradient descent optimization for F^* [80]. Therefore, every h_m is trained with a new training set $D = \{x_i, r_{mi}\}_{i=1}^{N}$, where r_{mi} represents the false residuals, and it is the difference between the output of an individual base classifier and the actual label [60]. The false residual is also called pseudo residuals, and it is computed as:

$$r_{mi} = [\frac{\delta L(y_i, F(x))}{\delta F(x)}]_{F(x)=F_{m-1}(x)} \tag{12}$$

Subsequently, the value of ρ_m is calculated by performing a line search optimization. Meanwhile, this algorithm could overfit if the iterative task is not regularized appropriately [78]. For certain loss functions, such as quadratic loss function, if h_m fits the false residuals perfectly, then in subsequent iteration the false residuals would be zero and the iteration ends early.

Furthermore, many regularization hyperparameters have been studied to optimize the GBDT's additive learning method. However, the intrinsic approach to regularize the GBDT is by using shrinkage to limit every gradient decent step $F_m(x) \ = F_{m-1}(x) + v\rho_m h_m(x)$, where v is normally assigned a value of 0.1 [82]. The gradient boosting procedure is summarized in Algorithm 2.

```
Algorithm 2 Gradient Boosting
Input: training data $S = (x_1, y_1), \ldots, (x_2, y_2), \ldots, (x_m, y_m)$
    A differential loss function $L(y, F(x))$.
    The number of iterations $T$.
    Procedure:
      1) Initialize the model with a constant value using
         $F_0 = argmin_\alpha \sum_{i=1}^{N} L(y_i, \alpha)$
      2) for $m = 1, \ldots, M$:
         (i) Calculate the false residuals
         $r_{mi} = [\frac{\delta L(y_i,F(x))}{\delta F(x)}]_{F(x)=F_{m-1}(x)}$ for $i = 1, \ldots, n$ (ii)
         Train a base learner using the training set
         $D = x_i, r_{mi}{}_{i=1}^{N}$
         (iii) Obtain $\rho_m$ by performing the line search optimization:
         $(\rho_m h_m(x)) = argmin_{\rho,h} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \rho h(x_i))$
         (iv) Update the model:
         $F_m(x) = F_{m-1}(x) \ + \rho_m h_m(x)$
    end for
Output: Return the final model $F_m(x)$.
```

The main advantage of gradient boosting is that, like other boosting algorithms, it can learn complex patterns from the input data since it is trained to correct the errors of the previous model. However, a model built using this algorithm can overfit and model noise if the input data is noisy [79], [83]. This algorithm is optimal for applications with small datasets [84].

3) XGBoost

The XGBoost algorithm is a decision tree-based ensemble that employs the gradient boosting framework. It is a scalable and highly accurate algorithm used for classification and regression applications. The XGBoost has recently dominated the applied machine learning domain and has won several Kaggle competitions. It was developed in 2016 by Chen and Guestrin [70], having several advancements compared to the conventional gradient boosting algorithm. Unlike the gradient boosting, the XGBoost loss function contains a regularization term that prevents overfitting [48]:

$$L_M(F(x_i)) = \sum_{i=1}^{n} L(y_i, F(x_i)) + \sum_{m=1}^{M} \Omega(h_m) \tag{13}$$

where $F(x_i)$ represents the prediction on the $i - th$ instance at the $M - th$ iteration, $L(*)$ represents a loss function that computes the differences between the predicted class and the actual class of the target variable. Meanwhile, $\Omega(h_m)$ denotes the regularization term, and it is formulated as:

$$\Omega(h) \ = \gamma T + \frac{1}{2}\lambda||\omega||^2 \tag{14}$$

where Υ represents the complexity parameter, and it controls the minimum loss reduction gain required for splitting an internal node. Assigning a high value to Υ leads to simpler trees. Meanwhile, T represents the number of leaves in the tree, λ is a penalty parameter, and ω denotes the output of the leaf nodes. Meanwhile, unlike the first-order derivative in GBDT, a second-order Taylor approximation of the objective function is employed in the XGBoost. Therefore, Equation 13 is transformed thus:

$$L_M \approx \sum_{i=1}^{n}[g_i f_m(x_i) + \frac{1}{2}h_i f_m^2(x_i)] + \Omega(h_m) \tag{15}$$

where g_i and h_i denote the first and second derivatives of the loss function. Assuming I_j represents the samples in leaf node j, then the final loss value is calculated by summing the loss values of the various leaf nodes. Hence, the objection function is represented as:

$$L_M \ = \sum_{j=1}^{T}[(\sum_{i \in I_j} g_i)\omega_j + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda)\omega_j^2] + \gamma T \tag{16}$$

Essentially, the optimization is modified to a quadratic approximation of the objective function. Additionally, due to the introduction of the regularization term in the XGBoost, it is not susceptible to overfitting [85]. Like the gradient boosting algorithm, the XGBoost employs maximum tree depth, learning rate, and subsampling to prevent the model from overfitting.

Some advantage of using the XGBooost algorithm is that it requires minimal feature engineering, such as data normalization and feature scaling because the algorithm can handle such situations. Also, it is capable of handling missing values. This algorithm can output feature importance, which can be used to understand the input features better and also perform feature selection. The XGBoost is faster than most ML algorithms, can handle large datasets, and is not prone to overfitting. Furthermore, it often outperforms other ML algorithms, which is why it has won several Kaggle competitions. However, it has a few limitations, including its high number of hyperparameters, making it difficult to tune [86], [87].

4) LightGBM

The light gradient boosting machine (LightGBM) is an efficient implementation of the gradient boosting algorithm, and it was developed in 2017 by researchers at Microsoft [88]. It can be used for classification, ranking, and other ML problems. The LightGBM algorithm uses two novel methods, Gradient-based One-Sided Sampling (GOSS) and Exclusive Feature Bundling (EFB), ensuring the algorithm trains faster and achieves high accuracy. The GOSS technique is a modification of the gradient boosting technique, which takes into account the training instances that lead to a larger gradient, thereby making the learning process fast and reducing the computational complexity of the model.

Specifically, the GOSS technique involves excluding a considerable number of training examples with small gradients and uses only the remaining examples to compute the information gain [89]. The reason behind excluding samples with small gradients is that instances with large gradients are more useful in calculating the information gain (IG). Hence, the GOSS technique achieves excellent estimation of the IG with a reduced sample size [88]. Meanwhile, the EFB method performs a feature selection task by bundling sparse mutually exclusive attributes, thereby reducing the number of attributes [90]. Several attributes are almost exclusive in sparse feature space, i.e. they hardly take nonzero values simultaneously [91]. A typical example of exclusive features is One-hot encoded features. Furthermore, the EFB technique bundles such features to reduce the dimension of the feature matrix [92].

The main benefit of the LightGBM is that it is fast and mostly leads to a very efficient model. Secondly, it has a low memory consumption since it converts continuous values to discrete bins. Thirdly, it achieves much higher accuracy than most boosting methods, resulting from the introduction of GOSS and EFB techniques. Lastly, the LightGBM algorithm performs well when trained with large datasets, with a faster training time than the XGBoost algorithm [93]. In terms of disadvantages, the LightGBM can overfit small training datasets easily as it performs better with large datasets. Also, splitting the tree leaf-wise could result in overfitting because more complex trees are produced.

Meanwhile, the LightGBM has been applied for different classification problems, achieving excellent results [94], [95], [96], [97], and its procedure is presented in Algorithm 3. A detailed explanation of the LightGBM technique can be found in [88]. Also, a comprehensive mathematical overview of the LightGBM algorithm is presented in [98].

Algorithm 3 LightGBM

```
Input: training data $S = (x_1, y_1), \dots, (x_2, y_2), \dots, (x_n, y_n)$
    The loss function $L(y, \Theta(x))$
    The number of iterations $T$.
    The sampling ratio of large gradient data $a$, and the sampling ratio of small gradient data $b$.
    Procedure:
      1) Merge mutually exclusive features of $x_i$, $i = \{1, \dots, N\}$ using the EFB technique.
      2) Initialize $\Theta_0\ (x) = argmin_c \sum_i^N L(y_i, c)$
      3) for t=1,...,T:
         (i) Compute the absolute values of gradients:
         $r_i = [\frac{\delta L(y_i, \Theta(x_i))}{\delta \Theta(x_i)}]_{\Theta(x)=\Theta_{t-1}(x)}, i = \{1, \dots, N\}$
         (ii) Resample the dataset using the GOSS technique
         $topN = a \times len(D)$; $randN = b \times len(D)$;
         $sorted = GetSortedIndices(abs(r))$;
         $A = sorted[1 : topN]$; $B = RandomPick(sorted[topN : len(D)], randN)$;
         D' = A + B;
         (iii) Calculate information gain values
         $V_j(d) = \frac{1}{n}(\frac{(\sum_{x_i \in A_i} r_i + \frac{1-a}{b} \sum_{x_i \in B_i} r_i)^2}{n_i^j(d)} + \frac{(\sum_{x_i \in A_i} r_i + \frac{1-a}{b} \sum_{x_i \in B_i} r_i)^2}{n_i^j(d)})$
         (iv) Obtain a new decision tree $\Theta_t\ (x)'$ on set $D'$
         (v) Update $\Theta_t\ (x) = \Theta_{t-1}\ (x) + \Theta_t(x)'$
    end for
Output: $\hat{\theta}(x) = \Theta_T(x)$
```

5) CatBoost

The CatBoost algorithm is an implementation of gradient boosting proposed in 2017 by Prokhorenkova *et al.* [99]. The algorithm effectively handles categorical features during the training phase. A notable improvement in CatBoost is its ability to perform unbiased gradient estimation that reduces overfitting. Therefore, in order to estimate the gradient of each example at every boosting iteration, the CatBoost algorithm omits that example from being used to train the current model [100].

Another notable improvement in the CatBoost algorithm is how it automatically transforms categorical features into numerical ones. Categorical features contain a discrete set of values termed categories that are mostly not comparable. Hence, these features are not suitable for building decision trees in their present state. The categorical features are often converted to numerical features at the preprocessing stage, where they are replaced with numerical values. One-hot encoding is the most common method applied to low-cardinality categorical attributes, in which the original feature is replaced with a binary variable.

Furthermore, another method used to handle categorical attributes is the Greedy Target-based Statistics (Greedy TS) which replaces categorical features with an equivalent average label value [101]. Assuming we have a dataset $S = \{(X_i, Y_i)\}_{i=1...n}$, where $X_i = (x_{i,1}, \ldots, x_{i,m})$ represents a vector of m features, and $Y_i \in R$ is the label value [102]. In the Greedy TS method, the categorical features are replaced with the average label value of the entire training set [103]. Therefore, $x_{i,k}$ is replaced with $\frac{\sum_{j=1}^{n}[x_{j,k}=x_{i,k}].Y_j}{\sum_{j=1}^{n}[x_{j,k}=x_{i,k}]}$, where $[\cdot]$ is an indicator function, i.e., $[x_{j,k} = x_{i,k}]$ would be 1 if $x_{j,k} = x_{i,k}$ and 0 otherwise. Meanwhile, this method produces models that overfit. For instance, if there is one sample from the category $x_{(i,k)}$ in the dataset, the new numerical feature would be equal to the label value of this sample [102].

However, CatBoost applies an improved and more robust approach that does not lead to overfitting and ensures all the examples in the training set are used for training the model. This method involves performing a random permutation of the training set, and for every sample, the algorithm calculates the average label value for the sample with the same category value located before the given one in the permutation [102]. If $\sigma = (\sigma_1, \ldots, \sigma_n)$ is the permutation, then $x_{\sigma_p,k}$ is replaced with

$$\frac{\sum_{j=1}^{p-1}[x_{\sigma_j,k} = x_{\sigma_p,k}]Y_{\sigma_j} + a.p}{\sum_{j=1}^{p-1}[x_{\sigma_j,k} = x_{\sigma_p,k}]Y_{\sigma_j} + a} \tag{17}$$

where P is a prior value, and a is the weight of the prior value. Meanwhile, the parameter $a > 0$. Furthermore, adding the prior value and the prior weight in the CatBoost algorithm ensures the noise obtained from low-frequency categories is reduced [104]. Prokhorenkova *et al.* [99], in their pioneer CatBoost article, compared its performance with that of XGBoost and LightGBM, and they stated that CatBoost is less likely to overfit than XGBoost or LightGBM. They attributed the enhanced performance to the above method used by the CatBoost algorithm to encode categorical features. The CatBoost algorithm achieves excellent performance and outperforms most ML algorithms when the input is categorical data, and it inherently handles missing data. However, its performance can be poor if the parameters are not tuned effectively.

B. BAGGING

The bootstrap aggregating (bagging) was developed in 1994 by Breiman [105] to enhance the classification performance of ML models by combining the predictions from randomly generated training sets. The author argued that perturbing the learning set could lead to significant modifications in the obtained predictor; hence bagging can improve accuracy [105]. Meanwhile, diversity is obtained in bagging by creating bootstrapped replicas of the input data, where several subsets of the input data are picked randomly with replacements from the original training set. Therefore, the various training sets are seen as diverse and used to train multiple base learners of the same ML algorithm.

Basically, the bagging method involves splitting the training data for each base learner using random sampling to generate *b* different subsets used to train *b* base learners. The *b* base learners are then combined using majority voting to obtain a strong classifier [27]. The bagging procedure is shown in Algorithm 4. Random forest is a popular implementation of the bagging technique

Bagging enhances the performance of base learners more if the algorithm used in learning the model is unstable. An unstable algorithm significantly changes its generalization ability when slight modifications are made to its input. Bagging focuses more on reducing the variance in the ensemble members than the bias. Therefore, bagging performs optimally when the ensemble members have high variance and low bias. An example of an unstable algorithm is the decision tree; hence, bagged decision trees usually perform better than the single decision tree. Meanwhile, k-nearest Neighbor (KNN) and naïve Bayes are examples of stable algorithms, and bagging does not perform well with these algorithms as base learners [28].

A major advantage of bagging is that it efficiently decreases the variance without increasing bias. Other advantages of bagging include its ability to introduce diversity in the input data because of the bootstrapping approach. For large datasets, bagging has less computational time than most ML algorithms since it trains the model with a small sample size [20]. Meanwhile, a limitation of bagging is that it enhances the model's accuracy without regard for interpretability. For example, if only one tree were applied as the base learner, a suitable and easy-to-interpret tree diagram would have been obtained; hence, the interpretability is neglected since bagging uses many decision trees. Also, the selected features during training are not interpretable in bagging, so there could be situations where certain vital features are never used.

1) RANDOM FOREST

Random forest is an ensemble algorithm that applies the bagging technique to build multiple decision trees using bootstrapped samples. The bagging technique generates random samples with replacements from the input data and trains the decision trees from the samples [106]. The decision tree is the main component in the random forest algorithm [107]. Meanwhile, the algorithm was first developed in 1995 by Ho [108] using the random subspace method, and in 2001 Breiman [51] developed an extended version of the algorithm.

The random forest algorithm has been widely applied for numerous tasks [109], [110], [111] because it is easy to

Algorithm 4 Bagging

```
Input: training data $S = (x_1, y_1), \ldots, (x_2, y_2), \ldots, (x_n, y_n)$
    Base ML algorithm $L$
    The number of base learners $T$.
    Procedure:
    for $t = 1, \ldots, T$ :
        1) Generate a bootstrap sample $S_j$ from the input data $S$
        2) Fit a base learner $h_j$ using $S_j$, i.e. $h_j = L(S_j)$
    end for
Output: Combine the outputs of the base learners, $H(x) = mode(h_1(x), \ldots, h_T(x))$
```

implement, fast, and obtains excellent performance. Two essential aspects of the random forest algorithm include the development of multiple decision trees during training and the combination of their predictions using majority voting. Since decision trees are prone to overfitting, the voting approach minimises the random forest's chances to overfit [106]. The random forest algorithm follows the parallel ensemble learning block diagram in Figure 1, where the base learners are decision trees.

Furthermore, this algorithm uses bagging and feature randomness in building a forest of uncorrelated decision trees. Meanwhile, feature randomness is achieved using the random subspace method that ensures the features are randomly selected for training each decision tree in the forest [112]. The correlation between the decision trees that make up the forest is reduced since the trees are trained using a random feature subset rather than the whole feature set.

The random forest uses a random subset of k attributes, unlike the traditional bagging that uses all p attributes at each node of the trees. Also, the optimal partitioning rules for the nodes are chosen from the specific random subset only. Furthermore, the random forest algorithm ensures the trees are diverse and uncorrelated by using only a subset of predictors [113]. Furthermore, combining several uncorrelated trees reduces the model's variance, thereby enhancing the classification accuracy [105]. Studies have shown that the performance of random forest is superior to the performance of the initial bagging algorithm [114]. The random forest algorithm is presented in Algorithm 5.

A significant advantage of using this algorithm is its ability to solve the overfitting issue common in decision tree models. This is achieved through the random feature subset selection. Secondly, the random forest can handle missing data [115]. Additionally, random forests usually achieve excellent performance when the input data contains many features, i.e. high dimensional data [116]. Meanwhile, some limitations exist when using this algorithm; for example, more trees are needed to obtain a more accurate classification. However, too many trees would slow down the model training process. Also, as the number of decision trees increases, the random forest becomes slow in making predictions.

Algorithm 5 Random Forest Algorithm

```
Input: training data $S = (x_1, y_1), \ldots, (x_2, y_2), \ldots, (x_n, y_n)$,
    $p$ attributes and class variables.
    The number of trees $T$ in the forest
    The number of class labels $C$
    Procedure:
    for $t = 1, \ldots, T$ :
        1) Generate a bootstrap sample $S_j$ from the input data $S$
        2) Fit a base learner $h_j$ using $S_j$, and for a given node $n$,
           (A) Randomly select $k$ attributes. Usually $k \approx \sqrt{p}$ for classification tasks.
           (B) Compute the best split features using the randomly selected feature subset.
           (C) Split the node using the optimal split features obtained in step B.
           Repeat A-C until the stopping criteria are achieved.

        3) Repeat steps 1 and 2 for $T$ times to build a forest of $T$ trees.
    end for
Output: Combine the outputs of the various trees. For a
    given test sample $x$, the final predicted class label from
    the $T$ trees is:
    $H_T(x) = argmax_j \sum_{k=1}^{K} I(h_k(x) = j$, for $j = 1, \ldots, C$
```

C. STACKING

Stacked generalization (Stacking) is an ensemble learning framework that trains a separate ML algorithm to combine the predictions from two or more ensemble members. It was introduced in 1992 by Wolpert [117] to reduce the generalization error in machine learning problems. Stacking is useful in situations where several ML models are uniquely skilful on a particular task; then, the stacking approach would use a separate ML model to learn when to use the predictions from the various models [118]. The stacking framework is depicted in Figure 3.

Specifically, it involves building models using multiple base algorithms, called level-0 models, and a meta-learning algorithm that trains another model to combine the predictions from the base models. The meta-model is termed a level-1 model [33]. The core idea in stacking is that the level-0 base learners are trained using the training dataset and are provided with out-of-sample or unseen data; their predicted target labels on the unseen data, together with the actual labels, form the input and output pairs of a new dataset used to train the meta-learner [119].

Meta-learning is the part of machine learning where algorithms are trained using the output of other ML algorithms and make more accurate predictions given the predictions made by the other base classifiers [120]. As shown in Figure 3, the meta-learner is an integral part of the stacking framework because it trains the model that makes the final prediction. The meta-classifier learns how best to combine base learners' predictions [121]. The stacking method is powerful because it uses the strengths of multiple well-performing classifiers to make classifications that are superior to the individual models that make up the ensemble.

Furthermore, stacking uses different base algorithms and the same dataset to obtain models that are diverse and approach the predictive modelling problem differently. Unlike bagging, which mainly uses decision tree models trained on subsets of the input data, the stacked models use different algorithms and are trained on the same dataset. Also, unlike boosting, which sequentially trains models to correct the predictions of previous models, stacking uses a single model to learn how to optimally combine the predictions from the base learners. Meanwhile, the meta-model is usually simple as it learns from the predictions made by the level-0 models. Therefore, linear classifiers, such as logistic regression, are often used as meta-learners. However, in regression problems, linear regression is mainly used as the meta-classifier [122]. The stacking framework has the following steps:

- Step 1: Apply the selected algorithms to train the base models using the training data. The choice of the base algorithms depends on the specific user and the problem domain.
- Step 2: Use the output of the base models to create a new dataset. Specifically, the predicted target labels of the base models are taken as new features, while the actual target labels are taken as the target variables in the new dataset. For example, if each instance in S is $\{x_i, y_i\}$, an equivalent instance $\{\hat{x}_i, y_i\}$ is obtained in the new dataset, where $\hat{x}_i = \{h_1(x_i), h_2(x_2), \ldots, h_T(x_i)\}$.
- Step 3: Train the selected meta-learner using the new dataset.

After the meta-model is obtained, it is applied to combine the various base models. For an out-of-sample instance x, its predicted target outcome after applying the stacking framework is $\hat{h}(h_1(x), h_2(x) \ldots, h_T(x))$, where $\hat{h}$ denotes the level-1 classifier and $\{h_1, h_2, \ldots, h_T\}$ are the level-0 classifiers [123]. The stacking ensemble framework is summarized in Algorithm 6. Lastly, stacking is considered a framework more than an actual machine learning algorithm. It is not as popular as boosting and bagging in applied machine learning because it is quite difficult to implement and could easily lead to data leakage if not correctly done [118].

The main benefit of stacking is that stacked models usually utilize the ability of several well-performing algorithms for making classifications that are better than any of the individual algorithms used to build the ensemble. Usually, stacked models have high accuracies, a significant reason why they have been applied in and won several ML competitions [124], [125], [126]. Also, stacked models have improved diversity since different ML algorithms are used for training the base models. For example, using a factorization model such as matrix factorization and tree-based models (such as decision tree and random forest) as base learners could provide a good diversity because the former is trained quite differently from the latter. Meanwhile, the main limitation of stacked models is that they can have high computational time when the training dataset is large because the entire dataset is used to train each base classifier.

Algorithm 6 Stacking

```
Input: training data $S = (x_1, y_1), \ldots, (x_2, y_2), \ldots, (x_m, y_m)$
    The base learning algorithms $T$
    Procedure:
    Step 1: Train base learning models
    for $t = 1, \ldots, T$:
    Fit a base learner $h_t$ using $S$
    end for
    Step 2: Obtain a new dataset from $S$
    for $t = 1, \ldots, T$:
    Obtain a new dataset containing $\{\hat{x}_i, y_i\}$, where $\hat{x}_i = \{h_1(x_i), h_2(x_2), \ldots, h_T(x_i)\}$
    end for
    Step 3: Train the meta-learner $\hat{h}$ using the new dataset
    return $H(x) = \hat{h}(h_1(x), h_2(2), \ldots, h_T(x))$
Output: A stacked ensemble classifier $H$.
```

IV. ENSEMBLE LEARNING APPLICATIONS IN RECENT LITERATURE

Ensemble learning methods have obtained excellent performance in numerous applications, attracting much attention in many research fields. These methods are capable of enhancing the generalization ability of single classifiers. This section highlights some applications of ensemble methods that have dominated the field of applied machine learning in recent years: medical diagnosis, fraud detection, and sentiment analysis [127], [128], [129], [130]. Each of these application areas had problems that made it difficult for traditional ML algorithms to perform well. The problems include the class imbalance in medical and fraud detection datasets, the limited sample size in medical datasets, too many redundant attributes in medical and fraud detection data, and the challenge of decoding the ambiguity of human language in sentiment analysis.

A. MEDICAL DIAGNOSIS

Recently, there have been numerous advancements in the application of machine learning for diagnosing diseases, such as heart disease, hypertension, cancer, and diabetes. The early detection of diseases is crucial in effectively managing the progression of the disease. Electrocardiograms, computerized tomography (CT) scans, and other medical tests can detect diverse diseases. Still, the high cost of using such machines has made it difficult for people to access them, especially in developing countries [131]. Meanwhile, machine learning-based methods have been developed to overcome the challenges associated with the traditional methods [132]. However, there are specific challenges faced

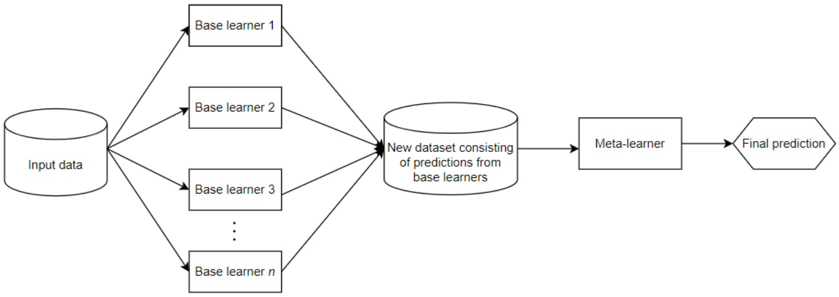


FIGURE 3. Block diagram of the Stacking framework.

by conventional ML algorithms when applied to medical datasets, including the imbalanced class problem and outliers in the dataset. These problems can affect the learning ability of conventional ML algorithms and reduce their prediction accuracy. Ensemble learning is one of the methods used to handle the class imbalance problem and outlier detection and elimination [133].

For example, Fitriyani *et al.* [134] developed an ensemble model for the early detection of hypertension and type 2 diabetes using the patient's risk factors information. The study used four different datasets: hypertension, prehypertension, type 2 diabetes, and chronic kidney disease datasets. Firstly, the study employed the SMOTE Tomek link (SMOTETomek) to create new datasets with even class distribution since all four datasets are imbalanced. Secondly, the study uses the isolation forest (iForest) algorithm [135] to detect and remove outliers in the datasets. Meanwhile, the iForest is an ensemble implementation that creates isolation trees using the given dataset. The isolation trees are repeatedly developed by splitting the training set until all the samples are isolated or the specified tree height is obtained. The experimental results show that the proposed approach achieved superior classification performance than other methods and prior research works. The proposed ensemble obtained 96.7%, 85.8%, 75.8%, and 100% accuracy when predicting type 2 diabetes, hypertension, prehypertension, and CKD, respectively.

Kazemi and Mirroshandel [136] proposed a novel ensemble method for the early detection of kidney stones, a common disease affecting people globally. The study employed several ML algorithms, including decision trees, naïve Bayes, and artificial neural networks (ANN), to learn the relationships between some biological features related to kidney stone disease. Furthermore, the study developed a novel method to combine the different classifiers. The combination method involved assigning weights to the different classifiers via a genetic algorithm (GA) based computation. Meanwhile, the data used in the research was obtained between 2012 and 2016 from 936 patients with kidney stone disease. The proposed ensemble achieved a classification accuracy of 97.1%. In addition, the ensemble model identified features that increase the risk of kidney stone disease, and these features include heart diseases, dehydration, gastric bypass surgery, prostate diseases, the consumption of calcium-based medications, and diuretic drugs.

Deep learning (DL) algorithms have recently received much attention due to their robust learning and generalization abilities. An *et al.* [137] proposed an ensemble approach coupled with deep learning techniques to predict Alzheimer's, a brain disorder associated with memory loss. Firstly, two sparse autoencoders (SAE) were employed for feature learning. Secondly, different ML algorithms were employed to develop several models from the learned data. A deep belief network (DBN) was utilized for training a meta-model that combines the predictions of the base models. The final deep ensemble was then applied to classify Alzheimer's disease using clinical data. The results show that the proposed method has accuracy 4% higher than six widely used ensemble algorithms.

There have been significant advancements in machine learning for medical diagnosis. However, many machine learning models only output the disease prediction or classification without explaining the fundamental decision-making process. Gu *et al.* [138] proposed using ensemble learning combined with case-based reasoning (CBR) for explainable breast cancer detection. The model performed a case-based explanation of the output predictions, thereby aiding clinicians in making better decisions. The XGBoost algorithm was used in building the predictive ensemble model, and the CBR provided an interpretation of the predictions. The proposed method is an improvement to recent ML research that has focused more on improving the accuracy rather than providing relevant explanations for the predictions.

Aljame *et al.* [139] applied ensemble learning to detect the novel coronavirus disease (COVID-19). The study aimed to detect the disease early using the XGBoost algorithm. Meanwhile, the dataset was first preprocessed to make it suitable for building the ML models. Hence, the missing values and outliers in the dataset were fixed using the k-nearest Neighbor (KNN) imputation technique and isolation forest, respectively. The dataset contained 5644 instances with 559 positive cases, indicating an imbalanced dataset. Therefore, the SMOTE technique was used to create a new dataset with an even class distribution. The XGBoost classifier trained with the preprocessed data obtained excellent classification results with an accuracy of 99.9%. The authors performed a comparative analysis with other well-performing studies in the literature, and the XGBoost ensemble showed superior performance.

Similarly, Abayomi-Alli *et al.* [140] used ensemble learning methods to build ML models that efficiently predict COVID-19. The methods include AdaBoost and random forest, and the AdaBoost classifier obtained the best performance with an accuracy of 99.3%. Furthermore, Mohammed *et al.* [141] used an ensemble of CNN classifiers to detect COVID-19, achieving an accuracy of 77.0%. Similarly, Ragab *et al.* [142] employed a deep learning ensemble to detect COVID-19. Before building the ensemble model, the dataset was preprocessed using the Gaussian filtering technique. The specific deep learning algorithms that made up the ensemble include recurrent neural network (RNN), gated recurrent unit (GRU), and long short-term memory (LSTM). The proposed ensemble classifier obtained a classification accuracy of 97.2%.

Mienye and Sun [143] utilized ensemble learning for heart disease prediction. The proposed approach involved using the synthetic minority oversampling technique-edited nearest Neighbor (SMOTE-ENN) to resample the dataset. Secondly, the recursive feature elimination (RFE) method was employed to select the most significant feature set for building the prediction model. Several classifiers were trained using the reduced feature set, and the XGBoost classifier obtained the best results with an accuracy of 95.6%. Furthermore, Mienye *et al.* [144] proposed an ensemble method to predict heart disease. The study applied the weighted ageing classifier ensemble (WAE) to combine the predictions from multiple classification and regression tree (CART) models. The proposed method achieved a classification accuracy of 93% when applied to the Cleveland heart disease dataset, outperforming similar techniques in the literature.

Gao *et al.* [145] studied the performance of the random forest ensemble classifier and other single ML algorithms for detecting heart disease. The single algorithms include SVM, naïve Bayes, decision tree, and KNN. The study used principal component analysis (PCA) and linear discriminant analysis (LDA) techniques to identify the most informative features in the dataset. The experimental results showed that the ensemble classifier outperformed the other algorithms, obtaining an accuracy of 98.6%. Similarly, Prakash and Karthikeyan [146] developed a heart disease prediction method using feature selection and ensemble learning. However, this study combined the genetic algorithm (GA) and LDA to achieve a robust feature selection model. The selected features were then employed to build a bagging-based predictive model using SVM, decision tree, and MLP as base learners. The proposed approach obtained an accuracy of 93.7%, outperforming other classifiers used for the performance comparison.

Velusamy and Ramasamy [147] proposed an ensemble classifier to detect heart disease. The ensemble was achieved by combining SVM, KNN, and random forest (RF). Several ensemble combination schemes were employed, including majority voting, weighted average voting (WAV), and average voting. The study employed the SMOTE technique to resample the data and obtain a balanced dataset for training the ML model. Also, the Boruta feature selection technique was used to select the optimal feature set. The experimental results indicated that the ensemble classifier obtained using the weighted average voting method got the highest accuracy of 100%. Furthermore, the statistical results showed the superior performance of the weighted average voting technique, which efficiently distinguished heart disease cases from healthy ones. Chicco and Jurman [148] applied ensemble learning to detect hepatitis. The study employed the random forest algorithm for feature selection and to build the prediction model, which was trained using electronic patient records containing 615 instances. Firstly, it identified alanine aminotransferase and aspartate aminotransferase enzymes as the most informative features in the dataset. The ensemble achieved a classification accuracy of 95.4%.

In another research, Ghiasi and Zendehboudi [149] used two decision tree-based ensemble algorithms to build breast cancer prediction models. The algorithms include the extra tree classifier (ETC) and the random forest algorithms, and the Wisconsin breast cancer dataset was used to train and test the models. The models achieved accuracies of 100%, outperforming the methods in previous literature, and the study concluded that both algorithms were effective at accurately diagnosing breast cancer. Similarly, Nanglia *et al.* [150] proposed a breast cancer detection model using an ensemble of multiple classifiers, including the decision tree, SVM, and KNN. The study employed a meta-classifier to combine the predictions of the ensemble members. The proposed ensemble achieved a classification accuracy of 78%, which was superior to the individual base models. Furthermore, the proposed ensemble model's performance was better than models developed using naïve Bayes, ANN, and logistic regression algorithms.

From the papers surveyed in this section, it is observed that ensemble methods have been widely utilized for diverse disease predictions and medical diagnoses, achieving excellent performance and outperforming traditional machine learning algorithms Table 1 presents a summary of the articles discussed in this section.

B. FRAUD DETECTION

Recently, due to technological advances and the fourth industrial revolution, most companies and organizations have embraced electronic commerce (e-commerce) systems, which has increased the use of credit cards and other electronic payment options. Fraudsters have targeted these e-commerce systems, hence, the need for efficient fraud detection systems to proactively detect and prevent fraud where possible [151]. Ensemble learning methods have been applied for fraud detection due to their robust performance and ability to handle imbalanced data. For example,

TABLE 1. Summary of the medical diagnosis papers.

Reference	Year	Ensemble Technique	Application	Accuracy(%)
Kazemi and Mirroshandel [136]	2020	GA-based ensemble classifier	Kidney-stones	97.1
Fitriyani et al. [134]	2019	iForest	Diabetes	96.7
An et al. [137]	2020	DBN-based DL ensemble	Alzheimer	-
Gu et al. [138]	2020	XGBoost and CBR	Breast cancer	-
Aljame et al. [139]	2020	XGBoost and KNN	COVID-19	99.9
Mienye et al. [144]	2020	WAE-based CART ensemble	Heart disease	93.0
Mohammed et al. [141]	2021	Ensemble of CNN models	COVID-19	77.0
Gao et al. [145]	2021	LDA-PCA and RF	Heart disease	98.6
Prakash and Karthikeyan [146]	2021	Bagging and GA-LDA	Heart disease	93.7
Velusamy and Ramasamy [147]	2021	WAV ensemble of SVM, KNN, and RF	Heart disease	100
Ghiasi and Zendehboudi [149]	2021	ETC and random forest	Breast cancer	100
Chicco and Jurman [148]	2021	Random forest	Hepatitis C	95.4
Nanglia et al. [150]	2021	Stacking based ensemble	COVID-19	97.2
Ragab et al. [142]	2022	Ensemble of RNN, GRU, and LSTM	COVID-19	97.2
Mienye and Sun [143]	2022	XGBoost	Heart disease	95.6
Abayomi-Alli et al. [140]	2022	AdaBoost	COVID-19	99.3

Xie *et al.* [152] proposed a credit card fraud detection system that integrates a KNN-based algorithm with a heterogeneous ensemble method to handle the class imbalance in the credit card dataset. The proposed approach achieved an excellent performance in classifying fraudulent and legitimate transactions.

Forough and Momtazi [153] proposed a fraud detection model using ensemble learning. The method applied a deep recurrent neural network (DRNN) and an ANN-based voting technique. The model was trained sequentially to identify fraudulent transactions using two real-world datasets. The DRNN ensured the time series inherent in the credit card data was modelled efficiently. The proposed method achieved precision values of 92.9% and 77.5% when trained with the European cardholders dataset and Brazilian credit dataset, respectively The performance of the proposed method was superior to the state-of-the-art methods used for credit card fraud detection.

Karthik *et al.* [154] proposed a hybrid ensemble method for credit card fraud detection. The proposed method combined boosting and bagging techniques to obtain a robust model. Mainly, the AdaBoost technique was used for feature engineering to obtain a suitable feature space. After that, learning models were developed using the extra tree classifier and random forest. The proposed AdaBoost combined with the extra tree classifier achieved an accuracy of 99.1%. In comparison, the proposed AdaBoost combined with the random forest classifier obtained an accuracy of 99.2%, outperforming other baseline models.

Alfaiz and Fati [155] developed credit card fraud detection models using the European cardholders dataset. The study used several ensemble-based methods, including random forest, XGBoost, LightGBM, CatBoost, and GBDT. These algorithms were combined with numerous resampling techniques to determine which combination obtains the best performance. The experimental results showed that the CatBoost with a KNN-based undersampling technique achieved the best performance with a sensitivity of 95.9% and AUC of 97.9%. Similarly, Ileberi *et al.* [156] applied several ML algorithms to detect credit card fraud using the European cardholders dataset. The study applied the genetic algorithm to select the most informative features in the dataset. The selected features were then used to train the ML algorithms, including a random forest ensemble, ANN, and logistic regression. The random forest classifier obtained the best performance with an accuracy of 99.9%.

Ileberi *et al.* [157] employed the AdaBoost algorithm to develop a credit card fraud detection framework. The framework involves a data resampling step using the SMOTE technique. Secondly, different algorithms were used individually as base learners in the AdaBoost implementation. The base learners included a decision tree, logistic regression, support vector machine, XGBoost, and random forest. The experimental results showed that the AdaBoost technique coupled with the selected algorithms achieved better classification performance compared to when the algorithms were used individually. Haider *et al.* [158] applied ensemble learning methods to detect advertisement (ad) fraud due to malicious ad displays. The ensemble methods used in the study include bagging and boosting techniques. Meanwhile, the dataset was first resampled using the SMOTE technique, and the most relevant features were selected using the information gain method. The experimental results indicated that the AdaBoost classifier obtained the best performance with an accuracy of 99%.

Esenogho *et al.* [26] proposed an ensemble learning-based credit card fraud detection method. The study aimed to develop a method capable of adapting to the dynamic nature of credit card transactions and solve the class imbalance in the dataset. The proposed method used the LSTM algorithm as the base classifier in the AdaBoost implementation. Meanwhile, the SMOTE-ENN technique was used to balance the dataset. Compared to other baseline models, the proposed ensemble learning method obtained superior performance, with a sensitivity of 99.6% and specificity of 99.8%.

Furthermore, Kewei *et al.* [159] proposed a deep learning-based ensemble for detecting online fraud. The method combined the binary cross-entropy (BCE) loss and

TABLE 2. Summary of the fraud detection papers.

Reference	Year	Ensemble technique	Application	Accuracy(%)
Haider et al. [158]	2018	AdaBoost	Advertisement fraud detection	99.0
Xie et al. [152]	2021	KNN Ensemble	Credit card fraud	-
Ileberi et al. [157]	2021	AdaBoost	Credit card fraud	-
Kewei et al. [159]	2021	Deep learning ensemble	Online fraud detection	95.8
Forough and Momtazi [153]	2021	DRNN and ANN-based ensemble	Credit card fraud	-
Karthik et al. [154]	2022	AdaBoost + Random forest	Credit card fraud	99.2
Alfaiz and Fati [155]	2022	CatBoost and KNN resampling	Credit card fraud	97.7
Ileberi et al. [156]	2022	Random forest	Credit card fraud	99.9
Esenogho et al. [26]	2022	LSTM based AdaBoost	Credit card fraud	-

Focal loss to obtain the proposed model's loss function. The proposed method ensures the model converges faster and learns effectively from the imbalanced dataset. The proposed deep learning ensemble achieved a classification accuracy of 95.8% and an AUC of 91.0%. Ensemble learning methods have also been applied for stock price manipulation [160] and recruitment fraud [161]. Table 2 presents a summary of the articles discussed in this section.

C. SENTIMENT ANALYSIS

Sentiment analysis (SA) (also known as opinion mining) is the application of natural language processing (NLP) to extract and analyze the emotional states and opinions of a writer or speaker [162]. Sentiment analysis has become a crucial research topic in machine learning with the rising number of social media applications and communities. Other than the traditional SA that detects the general sentiment of a specific text, aspect-based sentiment analysis (ABSA) identifies sentiment polarities (such as positive, negative, or neutral) of various aspects in a single sentence [163]. In the past, it was identified that the application of ensemble learning in sentiment analysis was limited [164]. However, there have recently been several applications of ensemble methods in sentiment analysis, and this section discusses some of those applications.

Basiri *et al.* [165] developed an ensemble classifier to analyze COVID-19-related tweets and Google searches to understand people's sentiments at various times and locations. The ensemble model was obtained by combining four deep learning algorithms and one conventional ML algorithm. The deep learning techniques include convolutional neural network (CNN), bidirectional gated recurrent network (BiGRU), fastText, and DistiBERT, while the conventional ML algorithm is the naïve Bayes support vector machines (NBSVM). The study employed the stacked generalization strategy to combine the five ensemble members. The research discovered that the coronavirus got people's attention in different countries at different times and intensities. The study also highlighted that the sentiments in the tweets are closely related to the news and happenings in the countries, such as the number of new infections, deaths and recoveries.

Similarly, Kandasamy *et al.* [166] proposed an ensemble deep learning model to analyze COVID-19-related sentiments and opinions among Twitter users. The study used the ensemble model to obtain better predictions than previous studies. Firstly, a feature extraction step was implemented using the N-gram stacked autoencoder neural network, and the output of this step was then used to build ML models using SVM, decision tree, KNN, and random forest algorithms. Secondly, the ML models were combined using mean and mode-based voting methods. The proposed deep learning-based feature extraction step integrated with the ensemble learning classification step was compared with other methods in recent literature, and the proposed method obtained superior performance with an accuracy of 87.8%.

Lin *et al.* [167] proposed an ensemble learning method to identify fake and harmful information in news reports. Usually, a news report is expected to be neutral, stating the facts, and should not contain too much of the writer's emotions and personal opinions. However, this is not always the case, as there are news reports that are written maliciously to achieve personal gains; hence, the study aimed to detect such news and classify them as fake. The study employed the bidirectional encoder representation from transformers (BERT) model using ensemble learning techniques with text sentiment classification. Specifically, the research employed bagging and stacking ensemble techniques to enhance the model's performance. The experimental results showed that the ensemble techniques enhanced the prediction of fake news with an accuracy of 99%.

AlGhamdi *et al.* [168] proposed an ensemble learning-based approach to perform aspect-oriented sentiment identification using datasets from multiple domains. The study employed the prior knowledge topic model algorithm and the stacking-based ensemble technique. From the experimental results, the proposed approach efficiently classify labels of the three review domains, i.e. Saudi airlines, restaurant, and movie, with a classification accuracy of 84.4%, 83.2%, and 84%, respectively. Meanwhile, the proposed approach achieved better performance compared to some baseline classifiers. Another aspect-based sentiment analysis model was developed by Zhang *et al.* [169] to predict customers' true consumption emotions and readiness to consume again using restaurant review data. Firstly, four conventional ML algorithms were used to develop single models. The algorithms include SVM, XGBoost, gradient boosting decision tree, and logistic regression. However, since a model developed using a single ML algorithm usually has inadequate diversity, the study employed the four algorithms as base learners

TABLE 3. Summary of the sentiment analysis papers.

Reference	Year	Ensemble technique	Application	Accuracy(%)
Basiri et al. [165]	2021	Deep learning ensemble	COVID-19 SA	-
Kandasamy et al. [166]	2021	Voting based ensemble	COVID-19 SA	87.8
Zhang et al. [169]	2021	Stacking based ensemble	Aspect-based sentiment analysis	-
Agarwal and Chowdary [171]	2021	Stacking based ensemble	Hate speech detection	-
Lin et al. [167]	2022	Bagging and stacking-based ensemble	Fake news detection	99.0
AlGhamdi et al. [168]	2022	Stacking based ensemble	Aspect-based sentiment analysis	84.0
Chen et al. [170]	2022	Stacking based ensemble	Humor detection	-
Abbasi et al. [172]	2022	Stacking based ensemble	Authorship detection	-
Karaoğlan and Fındık [173]	2022	Majority voting	Customer opinion extraction	-

in building an ensemble classifier, which outperformed the individual models.

Chen *et al.* [170] used the ensemble method to assess the degree of humor in text data. The research applied the stacking technique on 6 ML models. The models were developed using deep learning algorithms such as long short-term memory (LSTM), recurrent convolutional neural network (R-CNN), BERT, Text-CNN, BiLSTM, and feed-forward neural network (FFNN). The research employed the SemEval-2020 dataset, which is used to assess humor in edited news headlines. Firstly, the models were trained to obtain six separate text embeddings. Secondly, the text representations are stacked and fed into a meta-classifier which outputs the final prediction. Furthermore, the study concluded that applying the ensemble classifier led to better classification results than using the individual models. Other applications of ensemble learning in sentiment analysis include hate speech detection using stacking LSTM and RNN [171], authorship identification using a voting ensemble of multilayer perceptron, XGBoost, and random forest [172], and an opinion target extraction in restaurant review sentences using majority voting [173]. Table 3 presents a summary of the articles discussed in this section.

V. DISCUSSIONS AND FUTURE RESEARCH DIRECTIONS

Ensemble learning methods have recently obtained excellent performance in numerous applications and have attracted much attention, as seen in the preceding section. These methods are capable of enhancing the generalization ability of single classifiers. The number of ensemble learning methods presented in this paper is not intended to be all-inclusive. The methods discussed have been selected due to their popularity in diverse machine learning applications. The focus of this paper was to present a concise and straightforward overview of the selected ensemble learning algorithm, which is lacking in the existing literature.

In medical diagnosis, it is observed that random forest and XGBoost have been mostly used in the literature. Both methods offer reliable performances. In particular, XGBoost is quite robust and can handle missing values and class imbalance problems associated with medical datasets, possibly one of the reasons the XGBoost is popular in this domain. XGBoost handles missing values by default, but the scaleposweight hyperparameter is usually tuned for the algorithm to handle imbalanced data. Meanwhile, in fraud detection, the AdaBoost is the most popular. The AdaBoost algorithm is fast, simple, and easy to model with less need for hyperparameter tuning. A variety of algorithms can be used to train the base learner in AdaBoost. Therefore, the user can select an algorithm suitable for a specific task as the base estimator. For example, some research works used LSTM, a robust algorithm for time-series modelling, as the base estimator for credit card fraud detection, a task needing time-series modelling.

From the previous section, we can infer that stacking is mainly preferred for sentiment analysis. Several researchers have applied and proposed diverse stacking-based methods for different sentiment analysis problems. Firstly, the main challenge in sentiment analysis tasks is obtaining models that efficiently learn the data to achieve high performance. Stacking methods are mainly used because, unlike bagging and boosting, stacking employs different ML algorithms, which learn from the data differently. These algorithms approach the learning problem differently, leading to enhanced performance due to diversity in the ensemble model. Secondly, a meta-classifier learns how best to combine the base models, ensuring optimal performance.

As is evident in this paper and other research works, ensemble learning has evolved over the years and can be considered a mature subfield of machine learning compared to deep learning, transfer learning, and reinforcement learning. Furthermore, ensemble learning methods have been combined with deep learning for different applications, including human activity recognition [174], time series forecasting [175], disease prediction [137], [176], [177], wind speed prediction [178], and outlier detection [179]. Also, ensemble learning, deep learning, and transfer learning were combined in [180] to estimate the capacity of lithium-ion batteries.

The literature shows that the performance of deep learning techniques can be further improved when fused with some ensemble mechanism. For example, based on the notion that diverse convolutional neural network (CNN) methods learn different aspects of an image representation, in [181], an ensemble of CNNs was developed to achieve better feature extraction from medical images. The experimental results showed that the proposed approach obtained better feature extraction and accuracy than when a single CNN architecture was used. Even though much research has been carried out in ensemble deep learning and other areas, as seen in the previous section, many problems are still unexplored where ensemble methods can perform well. These unexplored areas are potential future research directions, and they are outlined below:

- Firstly, ensemble learning methods could potentially revolutionize applied machine learning by combining the vital functional elements of deep learning, transfer learning, and reinforcement learning to solve seemingly difficult and complex problems. Therefore, one future research direction is that ensemble learning would be instrumental in developing the next generation of start-of-the-art deep learning architectures. Also, we expect ensemble learning methods to play crucial roles in the advancements of transfer learning and reinforcement learning.
- Another future research direction is the application of ensemble learning to solve big data challenges. Big data has received significant attention in recent times. Deep learning has been a valuable tool for big data processing and modelling. Meanwhile, deep learning models are mostly complex and difficult to train compared to traditional ML models. Studying the suitability and benefit of ensemble methods in this fast-growing area could be worthwhile. Additionally, a few research works have already studied the application of ensemble learning in big data [182], [183], [184]. However, this area could benefit from more ensemble learning-based research.
- The use of ensemble techniques in clustering-based problems is a potential future research direction. The clustering domain has not benefitted from the vast number of ensemble algorithms. Recent studies have shown that conventional clustering methods, such as K-means clustering, tend to underperform when faced with high-dimensional data containing redundant features, sparse distribution, and outliers [185], [186]. In order to solve these problems, ensemble methods suitable for high-dimensional data could be studied and applied. Meanwhile, ensemble clustering has become a vital research area in cluster analysis. It aims to fuse two or more clustering models to obtain a superior performance compared to the single models [187], [188], [189]. Even though there are a few studies on ensemble clustering [190], more still needs to be done for it to become an established study area. Therefore, ensemble clustering is a recommended future research direction.

VI. CONCLUSION

Due to its robust learning ability, ensemble learning algorithms have been frequently applied in several classification and regression tasks in various domains such as medical diagnosis, fraud detection, sentiment analysis, and anomaly detection. This paper presents a brief but comprehensive overview of ensemble learning, from its early development to the recent state-of-the-art algorithms. This paper covers the three main categories of ensemble methods: bagging, boosting, and stacking. A focus is placed on the widely used ensemble algorithms, such as random forest, AdaBoost, XGBoost, LightGBM, and CatBoost. This paper will be crucial for machine learning researchers and practitioners who wish to understand ensemble learning and the popular ensemble algorithms.

REFERENCES

[1] B. van Giffen, D. Herhausen, and T. Fahse, "Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods," *J. Bus. Res.*, vol. 144, pp. 93–106, May 2022, doi: 10.1016/j.jbusres.2022.01.076.

[2] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron. Markets*, vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/s12525-021-00475-2.

[3] R. T. Aruleba, T. A. Adekiya, N. Ayawei, G. Obaido, K. Aruleba, I. D. Mienye, I. Aruleba, and B. Ogbuokiri, "COVID-19 diagnosis: A review of rapid antigen, RT-PCR and artificial intelligence methods," *Bioengineering*, vol. 9, no. 4, p. 153, Apr. 2022, doi: 10.3390/bioengineering9040153.

[4] A. T. Khan, X. Cao, S. Li, V. N. Katsikis, I. Brajevic, and P. S. Stanimirovic, "Fraud detection in publicly traded U.S firms using beetle antennae search: A machine learning approach," *Expert Syst. Appl.*, vol. 191, Apr. 2022, Art. no. 116148, doi: 10.1016/j.eswa.2021.116148.

[5] H. Chen, L. Wu, J. Chen, W. Lu, and J. Ding, "A comparative study of automated legal text classification using random forests and deep learning," *Inf. Process. Manage.*, vol. 59, no. 2, Mar. 2022, Art. no. 102798, doi: 10.1016/j.ipm.2021.102798.

[6] C. A. Graham, H. Shamkhalichenar, V. E. Browning, V. J. Byrd, Y. Liu, M. T. Gutierrez-Wing, N. Novelo, J.-W. Choi, and T. R. Tiersch, "A practical evaluation of machine learning for classification of ultrasound images of ovarian development in channel catfish (Ictalurus punctatus)," *Aquaculture*, vol. 552, Apr. 2022, Art. no. 738039, doi: 10.1016/j.aquaculture.2022.738039.

[7] A. Malekloo, E. Ozer, M. AlHamaydeh, and M. Girolami, "Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights," *Struct. Health Monitor.*, vol. 21, no. 4, pp. 1906–1955, Jul. 2022, doi: 10.1177/14759217211036880.

[8] X.-F. Song, Y. Zhang, D.-W. Gong, and X.-Z. Gao, "A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9573–9586, Sep. 2022, doi: 10.1109/TCYB.2021.3061152.

[9] W. Wang and D. Sun, "The improved AdaBoost algorithms for imbalanced data classification," *Inf. Sci.*, vol. 563, pp. 358–374, Jul. 2021, doi: 10.1016/j.ins.2021.03.042.

[10] I. D. Mienye and Y. Sun, "Performance analysis of cost-sensitive learning methods with application to imbalanced medical data," *Informat. Med. Unlocked*, vol. 25, Jan. 2021, Art. no. 100690, doi: 10.1016/j.imu.2021.100690.

[11] A. Mohammed and R. Kora, "An effective ensemble deep learning framework for text classification," *J. King Saud Univ. Comput. Inf. Sci.*, Nov. 2021, doi: 10.1016/j.jksuci.2021.11.001.

[12] P. Goel, R. Jain, A. Nayyar, S. Singhal, and M. Srivastava, "Sarcasm detection using deep learning and ensemble learning," *Multimedia Tools Appl.*, May 2022, doi: 10.1007/s11042-022-12930-z.

[13] J. Zhou, Z. Jiang, F.-L. Chung, and S. Wang, "Formulating ensemble learning of SVMs into a single SVM formulation by negative agreement learning," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 10, pp. 6015–6028, Oct. 2021, doi: 10.1109/TSMC.2019.2958647.

[14] I. D. Mienye, Y. Sun, and Z. Wang, "Improved predictive sparse decomposition method with densenet for prediction of lung cancer," *Int. J. Comput.*, vol. 19, no. 4, pp. 533–541, Dec. 2020, doi: 10.47839/ijc.19.4.1986.

[15] S. Mishra, K. Shaw, D. Mishra, S. Patil, K. Kotecha, S. Kumar, and S. Bajaj, "Improving the accuracy of ensemble machine learning classification models using a novel bit-fusion algorithm for healthcare AI systems," *Frontiers Public Health*, vol. 10, May 2022, Art. no. 858282, Accessed: Aug. 10, 2022, 10.3389/fpubh.2022.858282.

[16] Y. Sun, Z. Li, X. Li, and J. Zhang, "Classifier selection and ensemble model for multi-class imbalance learning in education grants prediction," *Appl. Artif. Intell.*, vol. 35, no. 4, pp. 290–303, Mar. 2021, doi: 10.1080/08839514.2021.1877481.

[17] G. Brown, ''Ensemble learning,'' in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA, USA: Springer, 2010, pp. 312–320, doi: 10.1007/978-0-387-30164-8_252.

[18] S. Doroudi, ''The bias-variance tradeoff: How data science can inform educational debates,'' *AERA Open*, vol. 6, no. 4, Oct. 2020, Art. no. 2332858420977208, doi: 10.1177/2332858420977208.

[19] L. Breiman, ''Arcing classifiers,'' *Ann. Stat.*, vol. 26, no. 3, pp. 801–824, 1998.

[20] S. Alelyani, ''Stable bagging feature selection on medical data,'' *J. Big Data*, vol. 8, no. 1, p. 11, Jan. 2021, doi: 10.1186/s40537-020-00385-8.

[21] K. A. Nguyen, W. Chen, B.-S. Lin, and U. Seeboonruang, ''Comparison of ensemble machine learning methods for soil erosion pin measurements,'' *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 1, p. 42, Jan. 2021, doi: 10.3390/ijgi10010042.

[22] M. H. D. M. Ribeiro and L. dos Santos Coelho, ''Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series,'' *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105837, doi: 10.1016/j.asoc.2019.105837.

[23] I. K. Nti, A. F. Adekoya, and B. A. Weyori, ''A comprehensive evaluation of ensemble learning for stock-market prediction,'' *J. Big Data*, vol. 7, no. 1, p. 20, Mar. 2020, doi: 10.1186/s40537-020-00299-5.

[24] K. Fawagreh, M. M. Gaber, and E. Elyan, ''Random forests: From early developments to recent advancements,'' *Syst. Sci. Control Eng.*, vol. 2, no. 1, pp. 602–609, 2014, doi: 10.1080/21642583.2014.956265.

[25] S. Mohammadi, Z. Narimani, M. Ashouri, R. Firouzi, and M. H. Karimi-Jafari, ''Ensemble learning from ensemble docking: Revisiting the optimum ensemble size problem,'' *Sci. Rep.*, vol. 12, no. 1, pp. 1–15, Jan. 2022, doi: 10.1038/s41598-021-04448-5.

[26] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, ''A neural network ensemble with feature engineering for improved credit card fraud detection,'' *IEEE Access*, vol. 10, pp. 16400–16407, 2022, doi: 10.1109/ACCESS.2022.3148298.

[27] Q.-F. Li and Z.-M. Song, ''High-performance concrete strength prediction based on ensemble learning,'' *Construct. Building Mater.*, vol. 324, Mar. 2022, Art. no. 126694, doi: 10.1016/j.conbuildmat.2022.126694.

[28] N. C. Oza and K. Tumer, ''Classifier ensembles: Select real-world applications,'' *Inf. Fusion*, vol. 9, pp. 4–20, Jan. 2008, doi: 10.1016/j.inffus.2007.07.002.

[29] C. Perales-González, F. Fernández-Navarro, M. Carbonero-Ruz, and J. Pérez-Rodríguez, ''Global negative correlation learning: A unified framework for global optimization of ensemble models,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 4031–4042, Aug. 2022, doi: 10.1109/TNNLS.2021.3055734.

[30] W. Li, Z. Wang, W. Sun, and S. Bahrami, ''An ensemble clustering framework based on hierarchical clustering ensemble selection and clusters clustering,'' *Cybern. Syst.*, pp. 1–26, May 2022, doi: 10.1080/01969722.2022.2073704.

[31] O. Sagi and L. Rokach, ''Ensemble learning: A survey,'' *WIREs Data Mining Knowl. Discovery*, vol. 8, no. 4, Jul. 2018, Art. no. e1249, doi: 10.1002/widm.1249.

[32] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, ''A survey on ensemble learning,'' *Frontiers Comput. Sci.*, vol. 14, no. 2, pp. 241–258, 2020, doi: 10.1007/s11704-019-8208-z.

[33] R. Polikar, ''Ensemble learning,'' in *Ensemble Machine Learning: Methods and Applications*, C. Zhang Y. Ma, Eds. Boston, MA, USA: Springer, 2012, pp. 1–34, doi: 10.1007/978-1-4419-9326-7_1.

[34] D. Guan, W. Yuan, Y.-K. Lee, K. Najeebullah, and M. K. Rasel, ''A review of ensemble learning based feature selection,'' *IETE Tech. Rev.*, vol. 31, no. 3, pp. 190–198, May 2014, doi: 10.1080/02564602.2014.906859.

[35] D. Che, Q. Liu, K. Rasheed, and X. Tao, ''Decision tree and ensemble learning algorithms with their applications in bioinformatics,'' in *Software Tools and Algorithms for Biological Systems*, H. R. Arabnia and Q.-N. Tran, Eds. New York, NY, USA: Springer, 2011, pp. 191–199, doi: 10.1007/978-1-4419-7046-6_19.

[36] Y. Ren, L. Zhang, and P. N. Suganthan, ''Ensemble classification and regression-recent developments, applications and future directions [review article],'' *IEEE Comput. Intell. Mag.*, vol. 11, no. 1, pp. 41–53, Feb. 2016, doi: 10.1109/MCI.2015.2471235.

[37] B. A. Tama and S. Lim, ''Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation,'' *Comput. Sci. Rev.*, vol. 39, Feb. 2021, Art. no. 100357, doi: 10.1016/j.cosrev.2020.100357.

[38] Y. Su, Y. Zhang, D. Ji, Y. Wang, and H. Wu, ''Ensemble learning for sentiment classification,'' in *Chinese Lexical Semantics*. Berlin, Germany: Springer, 2013, pp. 84–93, doi: 10.1007/978-3-642-36337-5_10.

[39] B. Krawczyk, L. L. Minkub, J. Gama, J. Stefanowski, and M. Woźniak, ''Ensemble learning for data stream analysis: A survey,'' *Inf. Fusion*, vol. 37, pp. 132–156, Sep. 2017, doi: 10.1016/j.inffus.2017.02.004.

[40] L. Rokach, ''Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography,'' *Comput. Statist. Data Anal.*, vol. 53, no. 12, pp. 4046–4072, 2009, doi: 10.1016/j.csda.2009.07.017.

[41] B. V. Dasarathy and B. V. Sheela, ''A composite classifier system design: Concepts and methodology,'' *Proc. IEEE*, vol. 67, no. 5, pp. 708–713, May 1979, doi: 10.1109/PROC.1979.11321.

[42] L. K. Hansen and P. Salamon, ''Neural network ensembles,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990, doi: 10.1109/34.58871.

[43] R. E. Schapire, ''The strength of weak learnability,'' *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990, doi: 10.1007/BF00116037.

[44] H. Liu, A. Gegov, and M. Cocea, ''Ensemble learning approaches,'' in *Rule Based Systems for Big Data: A Machine Learning Approach*, H. Liu, A. Gegov, and M. Cocea, Eds. Cham, Switzerland: Springer, 2016, pp. 63–73, doi: 10.1007/978-3-319-23696-4_6.

[45] T. Ravichandran, K. Gavahi, K. Ponnambalam, V. Burtea, and S. J. Mousavi, ''Ensemble-based machine learning approach for improved leak detection in water mains,'' *J. Hydroinformatics*, vol. 23, no. 2, pp. 307–323, Jan. 2021, doi: 10.2166/hydro.2021.093.

[46] M. Sabzevari, G. Martínez-Muñoz, and A. Suárez, ''Building heterogeneous ensembles by pooling homogeneous ensembles,'' *Int. J. Mach. Learn. Cybern.*, vol. 13, no. 2, pp. 551–558, Feb. 2022, doi: 10.1007/s13042-021-01442-1.

[47] Y. Bian, Y. Wang, Y. Yao, and H. Chen, ''Ensemble pruning based on objection maximization with a general distributed framework,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3766–3774, Sep. 2020, doi: 10.1109/TNNLS.2019.2945116.

[48] Y. Li and W. Chen, ''A comparative performance assessment of ensemble learning for credit scoring,'' *Mathematics*, vol. 8, no. 10, p. 1756, Oct. 2020, doi: 10.3390/math8101756.

[49] N. Li, Y. Yu, and Z.-H. Zhou, ''Diversity regularized ensemble pruning,'' in *Machine Learning and Knowledge Discovery in Databases*. Berlin, Germany: Springer, 2012, pp. 330–345, doi: 10.1007/978-3-642-33460-3_27.

[50] E. Alshdaifat, M. Al-Hassan, and A. Aloqaily, ''Effective heterogeneous ensemble classification: An alternative approach for selecting base classifiers,'' *ICT Exp.*, vol. 7, no. 3, pp. 342–349, Sep. 2021, doi: 10.1016/j.icte.2020.11.005.

[51] L. Breiman, ''Random forests,'' *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[52] R. E. Schapire, ''Boosting: Foundations and algorithms,'' *Kybernetes*, vol. 42, no. 1, pp. 164–166, Jan. 2013, doi: 10.1108/03684921311295547.

[53] I. H. Witten, E. Frank, and M. A. Hall, ''The explorer,'' in *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed., I. H. Witten, E. Frank, and M. A. Hall, Eds. Boston, MA, USA: Morgan Kaufmann, 2011, pp. 407–494, doi: 10.1016/B978-0-12-374856-0.00011-0.

[54] D. Ballabio, R. Todeschini, and V. Consonni, ''Recent advances in high-level fusion methods to classify multiple analytical chemical data,'' in *Data Handling in Science and Technology*, vol. 31, M. Cocchi, Ed. Amsterdam, The Netherlands: Elsevier, 2019, pp. 129–155, doi: 10.1016/B978-0-444-63984-4.00005-3.

[55] A. Dogan and D. Birant, ''A weighted majority voting ensemble approach for classification,'' in *Proc. 4th Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2019, pp. 1–6, doi: 10.1109/UBMK.2019.8907028.

[56] V. C. Osamor and A. F. Okezie, ''Enhancing the weighted voting ensemble algorithm for tuberculosis predictive diagnosis,'' *Sci. Rep.*, vol. 11, no. 1, pp. 1–11, Jul. 2021, doi: 10.1038/s41598-021-94347-6.

[57] N. Littlestone and M. K. Warmuth, ''The weighted majority algorithm,'' *Comput. Eng. Inf. Sci.*, vol. 108, no. 2, pp. 212–261, Feb. 1994, doi: 10.1006/inco.1994.1009.

[58] R. P. W. Duin, ''The combining classifier: To train or not to train?'' in *Proc. 16th Int. Conf. Pattern Recognit.*, vol. 2. Montreal, QC, Canada, Aug. 2002, pp. 765–770, doi: 10.1109/ICPR.2002.1048415.

[59] T. T. Nguyen, A. V. Luong, M. T. Dang, A. W.-C. Liew, and J. McCall, ''Ensemble selection based on classifier prediction confidence,'' *Pattern Recognit.*, vol. 100, Apr. 2020, Art. no. 107104, doi: 10.1016/j.patcog.2019.107104.

[60] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 18, doi: 10.1145/1015330.1015432.

[61] Q. Sun and B. Pfahringer, "Bagging ensemble selection," in *AI 2011: Advances in Artificial Intelligence*. Berlin, Germany: Springer, 2011, pp. 251–260, doi: 10.1007/978-3-642-25832-9_26.

[62] H. Guo, H. Liu, R. Li, C. Wu, Y. Guo, and M. Xu, "Margin & diversity based ordering ensemble pruning," *Neurocomputing*, vol. 275, pp. 237–246, Jan. 2018, doi: 10.1016/j.neucom.2017.06.052.

[63] A. H. R. Ko, R. Sabourin, and A. S. Britto, Jr., "From dynamic classifier selection to dynamic ensemble selection," *Pattern Recognit.*, vol. 41, no. 5, pp. 1718–1731, May 2008, doi: 10.1016/j.patcog.2007.10.015.

[64] W.-H. Hou, X.-K. Wang, H.-Y. Zhang, J.-Q. Wang, and L. Li, "A novel dynamic ensemble selection classifier for an imbalanced data set: An application for credit risk assessment," *Knowl.-Based Syst.*, vol. 208, Nov. 2020, Art. no. 106462, doi: 10.1016/j.knosys.2020.106462.

[65] R. M. O. Cruz, D. V. R. Oliveira, G. D. C. Cavalcanti, and R. Sabourin, "FIRE-DES++: Enhanced online pruning of base classifiers for dynamic ensemble selection," *Pattern Recognit.*, vol. 85, pp. 149–160, Jan. 2019, doi: 10.1016/j.patcog.2018.07.037.

[66] B. Jiao, Y. Guo, D. Gong, and Q. Chen, "Dynamic ensemble selection for imbalanced data streams with concept drift," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 22, 2022, doi: 10.1109/TNNLS.2022.3183120.

[67] P. Pérez-Gállego, A. Castaño, J. R. Quevedo, and J. J. del Coz, "Dynamic ensemble selection for quantification tasks," *Inf. Fusion*, vol. 45, pp. 1–15, Jan. 2019, doi: 10.1016/j.inffus.2018.01.001.

[68] M. Kearns and L. G. Valiant, "Crytographic limitations on learning Boolean formulae and finite automata," in *Proc. 21st Annu. ACM Symp. Theory Comput. (STOC)*, 1989, pp. 433–444, doi: 10.1145/73007.73049.

[69] Y. Freund and R. E. Schapire, "A short introduction to boosting," in *Proc. 16th Int. Joint Conf. Artif. Intell.*, 1999, pp. 1401–1406.

[70] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[71] I. D. Mienye, G. Obaido, K. Aruleba, and O. A. Dada, "Enhanced prediction of chronic kidney disease using feature selection and boosted classifiers," in *Intelligent Systems Design and Applications*. Cham, Switzerland, 2022, pp. 527–537, doi: 10.1007/978-3-030-96308-8_49.

[72] H. Jafarzadeh, M. Mahdianpari, E. Gill, F. Mohammadimanesh, and S. Homayouni, "Bagging and boosting ensemble classifiers for classification of multispectral, hyperspectral and PolSAR data: A comparative evaluation," *Remote Sens.*, vol. 13, no. 21, p. 4405, Nov. 2021, doi: 10.3390/rs13214405.

[73] J. R. Bertini and M. D. C. Nicoletti, "An iterative boosting-based ensemble for streaming data classification," *Inf. Fusion*, vol. 45, pp. 66–78, Jan. 2019, doi: 10.1016/j.inffus.2018.01.003.

[74] M. Kuhn and K. Johnson, *Applied Predictive Modeling*, 1st ed. New York, NY, USA: Springer, 2013.

[75] F. Wang, Z. Li, F. He, R. Wang, W. Yu, and F. Nie, "Feature learning viewpoint of AdaBoost and a new algorithm," *IEEE Access*, vol. 7, pp. 149890–149899, 2019, doi: 10.1109/ACCESS.2019.2947359.

[76] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997, doi: 10.1006/jcss.1997.1504.

[77] P. Wu and H. Zhao, "Some analysis and research of the AdaBoost algorithm," in *Intelligent Computing and Information Science*. Berlin, Germany: Springer, 2011, pp. 1–5, doi: 10.1007/978-3-642-18129-0_1.

[78] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[79] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers Neurorobot.*, vol. 7, p. 21, Dec. 2013. Accessed: Apr. 27, 2022, doi: 10.3389/fnbot.2013.00021.

[80] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artif. Intell. Rev.*, vol. 54, no. 3, pp. 1937–1967, Mar. 2021, doi: 10.1007/s10462-020-09896-5.

[81] J. H. Friedman, "Stochastic gradient boosting," *Comput. Statist. Data Anal.*, vol. 38, no. 4, pp. 367–378, 2002, doi: 10.1016/S0167-9473(01)00065-2.

[82] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 308–324, Sep. 2015, doi: 10.1016/j.trc.2015.02.019.

[83] B. Zhang, J. Ren, Y. Cheng, B. Wang, and Z. Wei, "Health data driven on continuous blood pressure prediction based on gradient boosting decision tree algorithm," *IEEE Access*, vol. 7, pp. 32423–32433, 2019, doi: 10.1109/ACCESS.2019.2902217.

[84] J. Jiang, R. Wang, M. Wang, K. Gao, D. D. Nguyen, and G.-W. Wei, "Boosting tree-assisted multitask deep learning for small scientific datasets," *J. Chem. Inf. Model.*, vol. 60, no. 3, pp. 1235–1244, Mar. 2020, doi: 10.1021/acs.jcim.9b01184.

[85] W. Liang, S. Luo, G. Zhao, and H. Wu, "Predicting hard rock pillar stability using GBDT, XGBoost, and LightGBM algorithms," *Mathematics*, vol. 8, no. 5, p. 765, May 2020, doi: 10.3390/math8050765.

[86] J. Nobre and R. F. Neves, "Combining principal component analysis, discrete wavelet transform and XGBoost to trade in the financial markets," *Expert Syst. Appl.*, vol. 125, pp. 181–194, Jul. 2019, doi: 10.1016/j.eswa.2019.01.083.

[87] B. Zhang, Y. Zhang, and X. Jiang, "Feature selection for global tropospheric ozone prediction based on the BO-XGBoost-RFE algorithm," *Sci. Rep.*, vol. 12, no. 1, pp. 1–10, Jun. 2022, doi: 10.1038/s41598-022-13498-2.

[88] G. Ke, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3149–3157, Accessed: May 9, 2022. [Online]. Available: https://papers.nips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[89] R. Wang, Y. Liu, X. Ye, Q. Tang, J. Gou, M. Huang, and Y. Wen, "Power system transient stability assessment based on Bayesian optimized LightGBM," in *Proc. IEEE 3rd Conf. Energy Internet Energy Syst. Integr. (EI2)*, Nov. 2019, pp. 263–268, doi: 10.1109/EI247390.2019.9062027.

[90] K. Neshatian and L. Varn, "Feature bundles and their effect on the performance of tree-based evolutionary classification and feature selection algorithms," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2019, pp. 1612–1619, doi: 10.1109/CEC.2019.8789951.

[91] Z. Huang and Z. Chen, "Comparison of different machine learning algorithms for predicting the SAGD production performance," *J. Petroleum Sci. Eng.*, vol. 202, Jul. 2021, Art. no. 108559, doi: 10.1016/j.petrol.2021.108559.

[92] Y. Chen, K. Liu, Y. Xie, and M. Hu, "Financial trading strategy system based on machine learning," *Math. Problems Eng.*, vol. 2020, pp. 1–13, Jul. 2020, doi: 10.1155/2020/3589198.

[93] D. A. McCarty, H. W. Kim, and H. K. Lee, "Evaluation of light gradient boosted machine learning technique in large scale land use and land cover classification," *Environments*, vol. 7, no. 10, p. 84, Oct. 2020, doi: 10.3390/environments7100084.

[94] L. Deng, J. Pan, X. Xu, W. Yang, C. Liu, and H. Liu, "PDRLGB: Precise DNA-binding residue prediction using a light gradient boosting machine," *BMC Bioinf.*, vol. 19, no. S19, p. 522, Dec. 2018, doi: 10.1186/s12859-018-2527-1.

[95] A. A. Taha and S. J. Malebary, "An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine," *IEEE Access*, vol. 8, pp. 25579–25587, 2020, doi: 10.1109/ACCESS.2020.2971354.

[96] Y. Ju, G. Sun, Q. Chen, M. Zhang, H. Zhu, and M. U. Rehman, "A model combining convolutional neural network and LightGBM algorithm for ultra-short-term wind power forecasting," *IEEE Access*, vol. 7, pp. 28309–28318, 2019, doi: 10.1109/ACCESS.2019.2901920.

[97] J. Song, G. Liu, J. Jiang, P. Zhang, and Y. Liang, "Prediction of protein–ATP binding residues based on ensemble of deep convolutional neural networks and LightGBM algorithm," *Int. J. Mol. Sci.*, vol. 22, no. 2, p. 939, Jan. 2021, doi: 10.3390/ijms22020939.

[98] Y. Wang and T. Wang, "Application of improved LightGBM model in blood glucose prediction," *Appl. Sci.*, vol. 10, no. 9, p. 3227, May 2020, doi: 10.3390/app10093227.

[99] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 6639–6649, Accessed: May 22, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html

[100] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class imbalanced data classification: An experimental review," *J. Big Data*, vol. 7, no. 1, p. 70, Sep. 2020, doi: 10.1186/s40537-020-00349-y.

[101] Y. Zhang, Z. Zhao, and J. Zheng, ''CatBoost: A new approach for estimating daily reference crop evapotranspiration in arid and semi-arid regions of northern China,'' *J. Hydrol.*, vol. 588, Sep. 2020, Art. no. 125087, doi: 10.1016/j.jhydrol.2020.125087.

[102] A. V. Dorogush, V. Ershov, and A. Gulin, ''CatBoost: Gradient boosting with categorical features support,'' 2018, *arXiv:1810.11363*.

[103] D. Niu, L. Diao, Z. Zang, H. Che, T. Zhang, and X. Chen, ''A machine-learning approach combining wavelet packet denoising with CatBoost for weather forecasting,'' *Atmosphere*, vol. 12, no. 12, p. 1618, Dec. 2021, doi: 10.3390/atmos12121618.

[104] C. Lu, S. Zhang, D. Xue, F. Xiao, and C. Liu, ''Improved estimation of coalbed methane content using the revised estimate of depth and CatBoost algorithm: A case study from southern Sichuan basin, China,'' *Comput. Geosci.*, vol. 158, Jan. 2022, Art. no. 104973, doi: 10.1016/j.cageo.2021.104973.

[105] L. Breiman, ''Bagging predictors,'' *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996, doi: 10.1007/BF00058655.

[106] P. D. Caie, N. Dimitriou, and O. Arandjelovic, ''Precision medicine in digital pathology via image analysis and machine learning,'' in *Artificial Intelligence and Deep Learning in Pathology*, S. Cohen, Ed. Amsterdam, The Netherlands: Elsevier, 2021, pp. 149–173, doi: 10.1016/B978-0-323-67538-3.00008-7.

[107] S. Suthaharan, ''Chapter 6–A cognitive random forest: An intra- and intercognitive computing for big data classification under cune condition,'' in *Handbook Statist.*, vol. 35, V. N. Gudivada, V. V. Raghavan, V. Govindaraju, and C. R. Rao, Eds. Amsterdam, The Netherlands: Elsevier, 2016, pp. 207–227, doi: 10.1016/bs.host.2016.07.006.

[108] T. K. Ho, ''Random decision forests,'' in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, vol. 1, Aug. 1995, pp. 278–282, doi: 10.1109/ICDAR.1995.598994.

[109] S. Janitza and R. Hornung, ''On the overestimation of random forest's out-of-bag error,'' *PLoS ONE*, vol. 13, no. 8, Aug. 2018, Art. no. e0201904, doi: 10.1371/journal.pone.0201904.

[110] M. Schonlau and R. Y. Zou, ''The random forest algorithm for statistical learning,'' *Stata J.*, vol. 20, no. 1, pp. 3–29, Mar. 2020, doi: 10.1177/1536867X20909688.

[111] J. Lee, J. Cai, F. Li, and Z. A. Vesoulis, ''Predicting mortality risk for preterm infants using random forest,'' *Sci. Rep.*, vol. 11, no. 1, pp. 1–9, Mar. 2021, doi: 10.1038/s41598-021-86748-4.

[112] N. Mrabah, M. Bouguessa, and R. Ksantini, ''Adversarial deep embedded clustering: On a better trade-off between feature randomness and feature drift,'' *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1603–1617, Apr. 2022, doi: 10.1109/TKDE.2020.2997772.

[113] S. Han and H. Kim, ''On the optimal size of candidate feature set in random forest,'' *Appl. Sci.*, vol. 9, no. 5, p. 898, Mar. 2019, doi: 10.3390/app9050898.

[114] N. Altman and M. Krzywinski, ''Ensemble methods: Bagging and random forests,'' *Nature Methods*, vol. 14, no. 10, pp. 933–934, Oct. 2017, doi: 10.1038/nmeth.4438.

[115] S. Hong and H. S. Lynn, ''Accuracy of random-forest-based imputation of missing data in the presence of non-normality, non-linearity, and interaction,'' *BMC Med. Res. Methodol.*, vol. 20, no. 1, p. 199, Jul. 2020, doi: 10.1186/s12874-020-01080-1.

[116] L. Capitaine, R. Genuer, and R. Thiébaut, ''Random forests for high-dimensional longitudinal data,'' *Stat. Methods Med. Res.*, vol. 30, no. 1, pp. 166–184, Jan. 2021, doi: 10.1177/0962280220946080.

[117] D. H. Wolpert, ''Stacked generalization,'' *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Jan. 1992, doi: 10.1016/S0893-6080(05)80023-1.

[118] I. H. Witten, E. Frank, and M. A. Hall, ''The knowledge flow interface,'' in *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed., I. H. Witten, E. Frank, and M. A. Hall, Eds. Boston, MA, USA: Morgan Kaufmann, 2011, pp. 495–503, doi: 10.1016/B978-0-12-374856-0.00012-2.

[119] M. Liang, T. Chang, B. An, X. Duan, L. Du, X. Wang, J. Miao, L. Xu, X. Gao, L. Zhang, and J. Li, ''A stacking ensemble learning framework for genomic prediction,'' *Frontiers Genet.*, vol. 12, Mar. 2021, Art. no. 600040, Accessed: May 2, 2022. [Online]. Available: https://www.frontiersin.org/article/10.3389/fgene.2021.600040

[120] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, ''Meta-learning in neural networks: A survey,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5149–5169, Sep. 2022, doi: 10.1109/TPAMI.2021.3079209.

[121] R. Vilalta and Y. Drissi, ''A perspective view and survey of meta-learning,'' *Artif. Intell. Rev.*, vol. 18, no. 2, pp. 77–95, 2002, doi: 10.1023/A:1019956318069.

[122] O. Miguel-Hurtado, R. Guest, S. V. Stevenage, G. J. Neil, and S. Black, ''Comparing machine learning classifiers and linear/logistic regression to explore the relationship between hand dimensions and demographic characteristics,'' *PLoS ONE*, vol. 11, no. 11, Nov. 2016, Art. no. e0165521, doi: 10.1371/journal.pone.0165521.

[123] Y. Li, J. Gao, Q. Li, and W. Fan, ''Ensemble learning,'' in *Data Classification*. London, U.K.: Chapman & Hall/CRC, 2014.

[124] A. Chatzimparmpas, R. M. Martins, K. Kucher, and A. Kerren, ''Empirical study: Visual analytics for comparing stacking to blending ensemble learning,'' in *Proc. 23rd Int. Conf. Control Syst. Comput. Sci. (CSCS)*, May 2021, pp. 1–8, doi: 10.1109/CSCS52396.2021.00008.

[125] C. Deotte, B. Liu, B. Schifferer, and G. Titericz, ''GPU accelerated boosted trees and deep neural networks for better recommender systems,'' in *Proc. RecSysChallenge, Recommender Syst. Challenge*, Oct. 2021, pp. 7–14, doi: 10.1145/3487572.3487605.

[126] Z. Ouyang, X. Sun, J. Chen, D. Yue, and T. Zhang, ''Multi-view stacking ensemble for power consumption anomaly detection in the context of industrial Internet of Things,'' *IEEE Access*, vol. 6, pp. 9623–9631, 2018, doi: 10.1109/ACCESS.2018.2805908.

[127] I. D. Mienye, Y. Sun, and Z. Wang, ''Improved sparse autoencoder based artificial neural network approach for prediction of heart disease,'' *Informat. Med. Unlocked*, vol. 18, Jan. 2020, Art. no. 100307, doi: 10.1016/j.imu.2020.100307.

[128] P. K. Jain, R. Pamula, and G. Srivastava, ''A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews,'' *Comput. Sci. Rev.*, vol. 41, Aug. 2021, Art. no. 100413, doi: 10.1016/j.cosrev.2021.100413.

[129] M. F. Ak, ''A comparative analysis of breast cancer detection and diagnosis using data visualization and machine learning applications,'' *Healthcare*, vol. 8, no. 2, p. 111, Apr. 2020, doi: 10.3390/healthcare8020111.

[130] N. F. Ryman-Tubb, P. Krause, and W. Garn, ''How artificial intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark,'' *Eng. Appl. Artif. Intell.*, vol. 76, pp. 130–157, Nov. 2018, doi: 10.1016/j.engappai.2018.07.008.

[131] S. A. Ebiaredoh-Mienye, T. G. Swart, E. Esenogho, and I. D. Mienye, ''A machine learning method with filter-based feature selection for improved prediction of chronic kidney disease,'' *Bioengineering*, vol. 9, no. 8, p. 350, Jul. 2022, doi: 10.3390/bioengineering9080350.

[132] I. D. Mienye and Y. Sun, ''Improved heart disease prediction using particle swarm optimization based stacked sparse autoencoder,'' *Electronics*, vol. 10, no. 19, p. 2347, Sep. 2021, doi: 10.3390/electronics10192347.

[133] M. Juez-Gil, Á. Arnaiz-González, J. J. Rodríguez, and C. García-Osorio, ''Experimental evaluation of ensemble classifiers for imbalance in big data,'' *Appl. Soft Comput.*, vol. 108, Sep. 2021, Art. no. 107447, doi: 10.1016/j.asoc.2021.107447.

[134] N. L. Fitriyani, M. Syafrudin, G. Alfian, and J. Rhee, ''Development of disease prediction model based on ensemble learning approach for diabetes and hypertension,'' *IEEE Access*, vol. 7, pp. 144777–144789, 2019, doi: 10.1109/ACCESS.2019.2945129.

[135] F. T. Liu, K. M. Ting, and Z.-H. Zhou, ''Isolation forest,'' in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.

[136] Y. Kazemi and S. A. Mirroshandel, ''A novel method for predicting kidney stone type using ensemble learning,'' *Artif. Intell. Med.*, vol. 84, pp. 117–126, Jan. 2018, doi: 10.1016/j.artmed.2017.12.001.

[137] N. An, H. Ding, J. Yang, R. Au, and T. F. A. Ang, ''Deep ensemble learning for Alzheimer's disease classification,'' *J. Biomed. Informat.*, vol. 105, May 2020, Art. no. 103411, doi: 10.1016/j.jbi.2020.103411.

[138] D. Gu, K. Su, and H. Zhao, ''A case-based ensemble learning system for explainable breast cancer recurrence prediction,'' *Artif. Intell. Med.*, vol. 107, Jul. 2020, Art. no. 101858, doi: 10.1016/j.artmed.2020.101858.

[139] M. AlJame, I. Ahmad, A. Imtiaz, and A. Mohammed, ''Ensemble learning model for diagnosing COVID-19 from routine blood tests,'' *Informat. Med. Unlocked*, vol. 21, 2020, Art. no. 100449, doi: 10.1016/j.imu.2020.100449.

[140] O. O. Abayomi-Alli, R. Damaševičius, R. Maskeliūnas, and S. Misra, ''An ensemble learning model for COVID-19 detection from blood test samples,'' *Sensors*, vol. 22, no. 6, p. 2224, Mar. 2022, doi: 10.3390/s22062224.

[141] E. A. Mohammed, M. Keyhani, A. Sanati-Nezhad, S. H. Hejazi, and B. H. Far, "An ensemble learning approach to digital corona virus preliminary screening from cough sounds," *Sci. Rep.*, vol. 11, no. 1, pp. 1–11, Jul. 2021, doi: 10.1038/s41598-021-95042-2.

[142] M. Ragab, K. Eljaaly, N. A. Alhakamy, H. A. Alhadrami, A. A. Bahaddad, S. M. Abo-Dahab, and E. M. Khalil, "Deep ensemble model for COVID-19 diagnosis and classification using chest CT images," *Biology*, vol. 11, no. 1, p. 43, Dec. 2021, doi: 10.3390/biology11010043.

[143] I. D. Mienye and Y. Sun, "Effective feature selection for improved prediction of heart disease," in *Pan-African Artificial Intelligence and Smart Systems*. Cham, Switzerland: Springer, 2022, pp. 94–107, doi: 10.1007/978-3-030-93314-2_6.

[144] I. D. Mienye, Y. Sun, and Z. Wang, "An improved ensemble learning approach for the prediction of heart disease risk," *Informat. Med. Unlocked*, vol. 20, Jan. 2020, Art. no. 100402, doi: 10.1016/j.imu.2020.100402.

[145] X.-Y. Gao, A. A. Ali, H. S. Hassan, and E. M. Anwar, "Improving the accuracy for analyzing heart diseases prediction based on the ensemble method," *Complexity*, vol. 2021, pp. 1–10, Feb. 2021, doi: 10.1155/2021/6663455.

[146] V. J. Prakash and N. K. Karthikeyan, "Enhanced evolutionary feature selection and ensemble method for cardiovascular disease prediction," *Interdiscipl. Sci., Comput. Life Sci.*, vol. 13, no. 3, pp. 389–412, Sep. 2021, doi: 10.1007/s12539-021-00430-x.

[147] D. Velusamy and K. Ramasamy, "Ensemble of heterogeneous classifiers for diagnosis and prediction of coronary artery disease with reduced feature subset," *Comput. Methods Programs Biomed.*, vol. 198, Jan. 2021, Art. no. 105770, doi: 10.1016/j.cmpb.2020.105770.

[148] D. Chicco and G. Jurman, "An ensemble learning approach for enhanced classification of patients with hepatitis and cirrhosis," *IEEE Access*, vol. 9, pp. 24485–24498, 2021, doi: 10.1109/ACCESS.2021.3057196.

[149] M. M. Ghiasi and S. Zendehboudi, "Application of decision tree-based ensemble learning in the classification of breast cancer," *Comput. Biol. Med.*, vol. 128, Jan. 2021, Art. no. 104089, doi: 10.1016/j.compbiomed.2020.104089.

[150] S. Nanglia, M. Ahmad, F. A. Khan, and N. Z. Jhanjhi, "An enhanced predictive heterogeneous ensemble model for breast cancer prediction," *Biomed. Signal Process. Control*, vol. 72, Feb. 2022, Art. no. 103279, doi: 10.1016/j.bspc.2021.103279.

[151] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *J. Netw. Comput. Appl.*, vol. 68, pp. 90–113, Jun. 2016, doi: 10.1016/j.jnca.2016.04.007.

[152] Y. Xie, A. Li, L. Gao, and Z. Liu, "A heterogeneous ensemble learning model based on data distribution for credit card fraud detection," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–13, Jul. 2021, doi: 10.1155/2021/2531210.

[153] J. Forough and S. Momtazi, "Ensemble of deep sequential models for credit card fraud detection," *Appl. Soft Comput.*, vol. 99, Feb. 2021, Art. no. 106883, doi: 10.1016/j.asoc.2020.106883.

[154] V. S. S. Karthik, A. Mishra, and U. S. Reddy, "Credit card fraud detection by modelling behaviour pattern using hybrid ensemble model," *Arabian J. Sci. Eng.*, vol. 47, no. 2, pp. 1987–1997, Feb. 2022, doi: 10.1007/s13369-021-06147-9.

[155] N. S. Alfaiz and S. M. Fati, "Enhanced credit card fraud detection model using machine learning," *Electronics*, vol. 11, no. 4, p. 662, Feb. 2022, doi: 10.3390/electronics11040662.

[156] E. Ileberi, Y. Sun, and Z. Wang, "A machine learning based credit card fraud detection using the GA algorithm for feature selection," *J. Big Data*, vol. 9, no. 1, p. 24, Feb. 2022, doi: 10.1186/s40537-022-00573-8.

[157] E. Ileberi, Y. Sun, and Z. Wang, "Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost," *IEEE Access*, vol. 9, pp. 165286–165294, 2021, doi: 10.1109/ACCESS.2021.3134330.

[158] C. M. R. Haider, A. Iqbal, A. H. Rahman, and M. S. Rahman, "An ensemble learning based approach for impression fraud detection in mobile advertising," *J. Netw. Comput. Appl.*, vol. 112, pp. 126–141, Jun. 2018, doi: 10.1016/j.jnca.2018.02.021.

[159] X. Kewei, B. Peng, Y. Jiang, and T. Lu, "A hybrid deep learning model for online fraud detection," in *Proc. IEEE Int. Conf. Consum. Electron. Comput. Eng. (ICCECE)*, Jan. 2021, pp. 431–434, doi: 10.1109/ICCECE51280.2021.9342110.

[160] Q. Wang, W. Xu, X. Huang, and K. Yang, "Enhancing intraday stock price manipulation detection by leveraging recurrent neural networks with ensemble learning," *Neurocomputing*, vol. 347, pp. 46–58, Jun. 2019, doi: 10.1016/j.neucom.2019.03.006.

[161] S. Lal, R. Jiaswal, N. Sardana, A. Verma, A. Kaur, and R. Mourya, "ORFDetector: Ensemble learning based online recruitment fraud detection," in *Proc. 12th Int. Conf. Contemp. Comput. (IC)*, Aug. 2019, pp. 1–5, doi: 10.1109/IC3.2019.8844879.

[162] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowl.-Based Syst.*, vol. 226, Aug. 2021, Art. no. 107134, doi: 10.1016/j.knosys.2021.107134.

[163] B. Liang, H. Su, L. Gui, E. Cambria, and R. Xu, "Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks," *Knowl.-Based Syst.*, vol. 235, Jan. 2022, Art. no. 107643, doi: 10.1016/j.knosys.2021.107643.

[164] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, "Sentiment classification: The contribution of ensemble learning," *Decision Support Syst.*, vol. 57, pp. 77–93, Jan. 2014, doi: 10.1016/j.dss.2013.08.002.

[165] M. E. Basiri, S. Nemati, M. Abdar, S. Asadi, and U. R. Acharrya, "A novel fusion-based deep learning model for sentiment analysis of COVID-19 tweets," *Knowl.-Based Syst.*, vol. 228, Sep. 2021, Art. no. 107242, doi: 10.1016/j.knosys.2021.107242.

[166] V. Kandasamy, P. Trojovský, F. A. Machot, K. Kyamakya, N. Bacanin, S. Askar, and M. Abouhawwash, "Sentimental analysis of COVID-19 related messages in social networks by involving an N-Gram stacked autoencoder integrated in an ensemble learning scheme," *Sensors*, vol. 21, no. 22, p. 7582, Nov. 2021, doi: 10.3390/s21227582.

[167] S.-Y. Lin, Y.-C. Kung, and F.-Y. Leu, "Predictive intelligence in harmful news identification by BERT-based ensemble learning model with text sentiment analysis," *Inf. Process. Manage.*, vol. 59, no. 2, Mar. 2022, Art. no. 102872, doi: 10.1016/j.ipm.2022.102872.

[168] N. AlGhamdi, S. Khatoon, and M. Alshamari, "Multi-aspect oriented sentiment classification: Prior knowledge topic modelling and ensemble learning classifier approach," *Appl. Sci.*, vol. 12, no. 8, p. 4066, Apr. 2022, doi: 10.3390/app12084066.

[169] Y. Zhang, H. Lu, C. Jiang, X. Li, and X. Tian, "Aspect-based sentiment analysis of user reviews in 5G networks," *IEEE Netw.*, vol. 35, no. 4, pp. 228–233, Jul. 2021, doi: 10.1109/MNET.011.2000400.

[170] J. Chen, Z. Li, and S. Qin, "Ensemble learning for assessing degree of humor," in *Proc. Int. Conf. Big Data, Inf. Comput. Netw. (BDICN)*, Jan. 2022, pp. 492–498, doi: 10.1109/BDICN55575.2022.00096.

[171] S. Agarwal and C. R. Chowdary, "Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19," *Expert Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115632, doi: 10.1016/j.eswa.2021.115632.

[172] A. Abbasi, A. R. Javed, F. Iqbal, Z. Jalil, T. R. Gadekallu, and N. Kryvinska, "Authorship identification using ensemble learning," *Sci. Rep.*, vol. 12, no. 1, pp. 1–16, Jun. 2022, doi: 10.1038/s41598-022-13690-4.

[173] K. M. Karaoğlan and O. Fındık, "Extended rule-based opinion target extraction with a novel text pre-processing method and ensemble learning," *Appl. Soft Comput.*, vol. 118, Mar. 2022, Art. no. 108524, doi: 10.1016/j.asoc.2022.108524.

[174] V. B. Semwal, A. Gupta, and P. Lalwani, "An optimized hybrid deep learning model using ensemble learning approach for human walking activities recognition," *J. Supercomput.*, vol. 77, pp. 12256–12279, Apr. 2021, doi: 10.1007/s11227-021-03768-7.

[175] X. Qiu, L. Zhang, Y. Ren, P. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," in *Proc. IEEE Symp. Comput. Intell. Ensemble Learn. (CIEL)*, Dec. 2014, pp. 1–6, doi: 10.1109/CIEL.2014.7015739.

[176] S. M. Usman, S. Khalid, and S. Bashir, "A deep learning based ensemble learning method for epileptic seizure prediction," *Comput. Biol. Med.*, vol. 136, Sep. 2021, Art. no. 104710, doi: 10.1016/j.compbiomed.2021.104710.

[177] H.-I. Suk, S.-W. Lee, and D. Shen, "Deep ensemble learning of sparse regression models for brain disease diagnosis," *Med. Image Anal.*, vol. 37, pp. 101–113, Apr. 2017, doi: 10.1016/j.media.2017.01.008.

[178] H. Liu, C. Yu, H. Wu, Z. Duan, and G. Yan, "A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting," *Energy*, vol. 202, Jul. 2020, Art. no. 117794, doi: 10.1016/j.energy.2020.117794.

[179] D. Chakraborty, V. Narayanan, and A. Ghosh, "Integration of deep feature extraction and ensemble learning for outlier detection," *Pattern Recognit.*, vol. 89, pp. 161–171, May 2019, doi: 10.1016/j.patcog.2019.01.002.

[180] S. Shen, M. Sadoughi, M. Li, Z. Wang, and C. Hu, "Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries," *Appl. Energy*, vol. 260, Feb. 2020, Art. no. 114296, doi: 10.1016/j.apenergy.2019.114296.

[181] A. Kumar, J. Kim, D. Lyndon, M. Fulham, and D. Feng, "An ensemble of fine-tuned convolutional neural networks for medical image classification," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 1, pp. 31–40, Jan. 2017, doi: 10.1109/JBHI.2016.2635663.

[182] A. Galicia, R. Talavera-Llames, A. Troncoso, I. Koprinska, and F. Martínez-Álvarez, "Multi-step forecasting for big data time series based on ensemble learning," *Knowl.-Based Syst.*, vol. 163, pp. 830–841, Jan. 2019, doi: 10.1016/j.knosys.2018.10.009.

[183] D. Gupta and R. Rani, "Improving malware detection using big data and ensemble learning," *Comput. Electr. Eng.*, vol. 86, Sep. 2020, Art. no. 106729, doi: 10.1016/j.compeleceng.2020.106729.

[184] A. Abbasi, A. R. Javed, C. Chakraborty, J. Nebhen, W. Zehra, and Z. Jalil, "ElStream: An ensemble learning approach for concept drift detection in dynamic social big data stream learning," *IEEE Access*, vol. 9, pp. 66408–66419, 2021, doi: 10.1109/ACCESS.2021.3076264.

[185] Š. Brodinová, P. Filzmoser, T. Ortner, C. Breiteneder, and M. Rohm, "Robust and sparse k-means clustering for high-dimensional data," *Adv. Data Anal. Classification*, vol. 13, pp. 905–932, Mar. 2019, doi: 10.1007/s11634-019-00356-9.

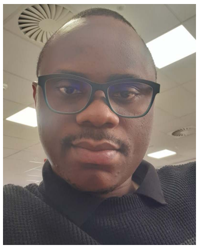
[186] A. Ultsch and J. Lötsch, "Machine-learned cluster identification in high-dimensional data," *J. Biomed. Informat.*, vol. 66, pp. 95–104, Feb. 2017, doi: 10.1016/j.jbi.2016.12.011.

[187] T. Alqurashi and W. Wang, "Clustering ensemble method," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 6, pp. 1227–1246, 2019, doi: 10.1007/s13042-017-0756-7.

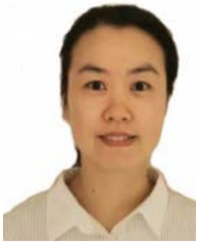
[188] I. Khan, Z. Luo, A. K. Shaikh, and R. Hedjam, "Ensemble clustering using extended fuzzy k-means for cancer data analysis," *Expert Syst. Appl.*, vol. 172, Jun. 2021, Art. no. 114622, doi: 10.1016/j.eswa.2021.114622.

[189] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "Marginalized multiview ensemble clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 600–611, Feb. 2020, doi: 10.1109/TNNLS.2019.2906867.

[190] J. Yan and W. Liu, "An ensemble clustering approach (consensus clustering) for high-dimensional data," *Secur. Commun. Netw.*, vol. 2022, pp. 1–9, May 2022, doi: 10.1155/2022/5629710.



IBOMOIYE DOMOR MIENYE (Member, IEEE) received the B.Eng. degree in electrical and electronic engineering and the M.Sc. degree *(cum laude)* in computer systems engineering from the University of East London, in 2012 and 2014, respectively, and the Ph.D. degree in electrical and electronic engineering from the University of Johannesburg, South Africa. He is currently a Postdoctoral Research Fellow with the Department of Electrical and Electronic Engineering Science, University of Johannesburg. His research interests include machine learning and deep learning for finance and healthcare applications.



YANXIA SUN (Senior Member, IEEE) received the D.Tech. degree in electrical engineering from the Tshwane University of Technology, South Africa, and the Ph.D. degree in computer science from University Paris-EST, France, in 2012. She is currently working as a Professor with the Department of Electrical and Electronic Engineering Science, University of Johannesburg, South Africa. Her research interests include renewable energy, evolutionary optimization, neural networks, non-linear dynamics, and control systems.