Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 8608630, 12 pages
https://doi.org/10.1155/2021/8608630

Research Article

Ensemble-Based Machine Learning for Predicting Sudden Human Fall Using Health Data

Utkarsh Saxena,[1] Soumen Moulik,[1] Soumya Ranjan Nayak,[2] Thomas Hanne,[3] and Diptendu Sinha Roy[1]

[1] *Department of Computer Science and Engineering, National Institute of Technology Meghalaya, Shillong, India*
[2] *Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India*
[3] *University of Applied Sciences and Arts Northwestern Switzerland, Olten, Switzerland*

Correspondence should be addressed to Thomas Hanne; thomas.hanne@fhnw.ch

Received 26 August 2021; Revised 28 September 2021; Accepted 28 October 2021; Published 23 November 2021

Academic Editor: Jude Hemanth

We attempt to predict the accidental fall of human beings due to sudden abnormal changes in their health parameters such as blood pressure, heart rate, and sugar level. In medical terminology, this problem is known as *Syncope*. The primary motivation is to prevent such falls by predicting abnormal changes in these health parameters that might trigger a sudden fall. We apply various machine learning algorithms such as logistic regression, a decision tree classifier, a random forest classifier, *K*-Nearest Neighbours (KNN), a support vector machine, and a naive Bayes classifier on a relevant dataset and verify our results with the cross-validation method. We observe that the KNN algorithm provides the best accuracy in predicting such a fall. However, the accuracy results of some other algorithms are also very close. Thus, we move one step further and propose an ensemble model, Majority Voting, which aggregates the prediction results of multiple machine learning algorithms and finally indicates the probability of a fall that corresponds to a particular human being. The proposed ensemble algorithm yields 87.42% accuracy, which is greater than the accuracy provided by the KNN algorithm.

1. Introduction

It has been studied that, as the age of a person increases, the chance of an accidental fall increases. More than one-third of the elderly suffer from accidental falls, and when this happens, they develop a sense of fear and loss of independence. Commonly, accidental falls occur indoors during locomotion, such as on stairs and in bathrooms. According to a global survey report published by the World Health Organization (WHO), the percentage of falls ranges between 28 and 35 for the age group of 65 years and from 32% to 42% for the age group of 70 and above [1]. The most common causes of such falls include age-related biological changes. These falls can lead to very severe injuries or even death. It has also been found that 50% of hospitalization cases for people above 65 years are due to some injuries.

Furthermore, as estimated by a report of the United Nations, the current world population is around 7.6 billion, out of which 962 million (13 percent of the world population) are elderly persons [2]. We live in a time when the population of the elderly in the world is on the verge of a steep rise. In 2019, the total global number of persons aged 65 years or more was 703 million, 9% of the total population. In the year 1990, this share was 6%. As predicted by the United Nations Department of Economic and Social Affairs in their report “World Population Ageing 2019: Highlights,” one in six people in the world will be aged 65 years or older by 2050, in place of the present ratio of one in eleven. Human life expectancy is in a growth phase. In the 1980s, the probability of surpassing the age of 65 was less than 50%. However, it has become more than 90% in countries with very high life expectancy. At present, older people make up more than a fifth of the population in 17 countries. As predicted, this will be the case by the end of this century for 155 countries, covering a majority (approximately 61%) of the world's population [3]. In another report titled "World Population Prospects 2019: Highlights" created by the same department, it was revealed that, in 2018, for the first time in history, persons aged 65 years or older outnumbered children under the age of five worldwide. By 2050, it will be more than twice the latter and will globally surpass the number of adolescents and youth aged 15 to 24 years [3]. The elderly population will grow to 18.6 percent in the upcoming 30 years, and it will grow at a faster rate than that of the young at an average rate of 3 percent every year. Therefore, detection and, most importantly, precise prediction of accidental falls are a timely need, especially for elderly persons.

It has been found that the major cause of the accidental falls, especially for elderly persons, is a sudden loss of consciousness or fainting due to the lack of blood flow to the brain. This means such accidental falls have a direct relationship with the underlying physiological condition of a human being. The proposed work exploits this relationship to predict and prevent human falls. The specific contributions of the proposed work are summarised below.

(i) Establishing a relationship between a person's health parameters' values and activity status and identifying the health conditions that indicate a fall.

(ii) Analysing the above-mentioned relationship using popular machine learning algorithms to find out the performance of these algorithms in terms of various metrics, most importantly accuracy.

(iii) Keeping in mind the future possibility of end-to-end system development, suggesting an ensemble method or algorithm which will aggregate the decisions of more than one machine learning algorithm, and finally notifying the percentage probability of a human being's fall.

2. Related Works

The existing works in this domain can be roughly divided into two groups: works based on fall detection and works based on fall prediction. Some important works from both categories are discussed below.

2.1. Works Related to Fall Detection. The works based on fall detection can be further divided into groups: (1) fall detection with wearable devices, (2) fall detection based on ambient sensing, and (3) fall detection based on camera. In case of wearable devices, accelerometers and gyroscopes are the most commonly used sensors. For example, the digital triaxial accelerometer ADXL345, tied to the waist of their users, was used by Wu et al. [4]. The recognition parameters here were derived acceleration and angular velocity. The authors used a threshold-based approach to detect a fall. Li et al. [5] found some problems associated with the use of accelerometers when recording vertical accelerations as many rigorous daily activities, such as sitting and lying down, can also produce similar results to those of fall. Later, several authors also accepted the fact that their models could not perform well during daily activities such as jumping into bed and falling against an object or wall in a seated idle position. In case of ambient-based fall detection techniques, the authors used several sensors to collect surrounding data when an employed user is in close proximity to a placed sensor. Alwan et al. [6] used the floor to install multiple vibration-based sensors to capture the event of a fall on the floor. Johnson [7] and Sixsmith et al. [8] used infrared sensors to detect the daily human activities and installed these sensors on the surrounding walls. Moreover, home assistance is heavily facilitated by the use of camera-based systems which are less intrusive as the cameras are mainly installed on the roof and are not needed to be worn by the end user. Camera-based fall detection methods use several characteristics, such as user inactivity inside the room, 3D head motion, and change in body shape, to decide whether a fall has occurred or not.

Moulik and Majumdar [9] used a combination of wearable sensors and ambient sensors to detect a fall. A Fuzzy Inference System (FIS) is used to conclude their result regarding a fall. The work done by the authors shows an average reduction of 16% in generating false alarms compared to the single sensor threshold-based methods already in use. This was a basic idea of existing approaches to determining that a fall has taken place and to suggest measures to reduce the aftereffects of a fall. The authors have tried to enable every individual user to create alarms and let their family and loved ones know they need them.

Apart from the above-mentioned works, a few interesting works are done using different data analysis methods on publicly available datasets. For example, Waheed et al. [10] proposed a fault-tolerant fall detection system using BI-LSTM to classify different falls on two public datasets. Their model is tolerant towards the missing values as they are adding Missing Completely at Random (MCAR) missingness in the dataset and performing 3D sampling on them. The authors added the missingness of 20%, 30%, and 40% in the datasets as, in a real-life scenario, it often happens that the dedicated wearable module or any data acquisition module fails to capture all the data simultaneously. Hence, results regarding many sensor-based features are missing. Because of the missing data, different models such as deep learning and other machine learning models are not able to perform well and provide efficient results. To mitigate this problem, the authors employed the Bi-LSTM deep learning model because it can use its memory to learn from the previous data. For future results, this model can easily predict the outcome, which contributes to better accuracy. The authors managed to achieve the highest accuracy of 97.21% and 97.41% using two public datasets and their proposed deep learning architecture.

Zurbuchen et al. [11] compared the performance of different machine learning algorithms with different publicly available datasets. They followed the general framework of applying machine learning algorithms using data preprocessing and feature extraction in the first phase. Later, with the help of five machine learning algorithms, they detected the falls and measured the performance in terms of accuracy, specificity, sensitivity, and F1-score. The authors also calculated the different accuracies of all the employed algorithms at different sampling rates of sensors values. Their experiment on the sampling rate finds out that the higher sampling rate sensors values give better results than the lower ones.

Iazzi et al. [12] proposed a fall detection system that uses different routine postures of a person. The authors' main aim is to classify the abnormal postures, often produced by accidental falls other than the general ones. They also employed two public datasets to implement their experiments and used a support vector machine (SVM) to detect falls. They applied feature extraction by converting the different images into a structural feature set. With the help of the SVM classifier, they detected different falls and other daily living activities. Finally, they achieved the highest accuracy of 99.78%, which outperforms the previous benchmarks.

Villaverde et al. [13] proposed a game theory-based efficient data collection for syncope and forward fall. They emphasised that the usual way of data collection for falls using a dummy is not an efficient option since it will lack natural human behaviour and cost more because of its excessive man-hour requirements. To overcome this problem, they simulated the real falls in the video game environment using UNITY software. They applied complex physics to two types of falls in unity and fathered the data from there. This helps capturing noise- and error-free data and helps in visualising the falls in real time. The authors' simulation employed a three-dimensional accelerometer sensor on a 3D model using NVidia's Physx System Software. Utilising this technique of data acquisition, they managed to maintain an accuracy above 98% on all the falls using the random forest classifier.

2.2. Works Related to Fall Prediction. Although much research for fall detection has been done and is still ongoing, one can observe a scarcity of quality works for fall prediction. We found two works where the authors used a dataset, provided by the Armed Forces Institute of Cardiology (AFIC) and the National Institute of Heart Disease (NIHD). This dataset is based on real case studies and consists of only 75 data points. Guftar et al. [14] used data mining techniques on this dataset, such as the K-means algorithm and the ELBOW technique, to determine the optimal value of k. The authors investigated the correlations between the different attributes in the dataset, such as age, gender, smoker, chest pain, and dizziness, to extract useful patterns and features that effectively support syncope prediction. Their experiment was carried out using four major techniques: K-means, K-means (fast), K-medoids, and X-means. The authors compared other algorithms with K-means and concluded that K-means is fast and X-means performed comparatively similar to K-means which were far better than K-medoid.

Guftar et al. [15] presented a different perspective and approach to predicting human fall. They considered the same dataset and used association rules to retrieve the user's chances of falling due to health abnormality. An association rule base was generated, denoted as a knowledge base. This knowledge is used to derive the outcome based upon the person's health parameter readings. The chances of human fall are broadly divided into low, medium, and high categories. Based on the user's health parameter readings such as blood pressure (BP), heart rate, dizziness, age, medications, family history, and other similar attributes, the authors classified the users as persons having high, medium, and low probabilities of a sudden fall. The authors also divided the dataset into training and the test data in the ratio of 65 : 35. In this work, the authors calculated the precision, recall, and accuracy scores as all 100%.

Synthesis: to summarise all the points and concepts studied by the authors in their respective works, we have found that, in the past, not much focus was put on predicting human falls triggered by health conditions. The existing works mainly kept their focuses on fall detection, and in this case, the works are also limited in many aspects. Firstly, the ambient sensor-based and camera-based fall detection methods [6–8] are confined within a room or ambiance, and therefore not ubiquitous in nature. Furthermore, camera-based methods are computationally expensive.

Secondly, in the case of wearable sensor-based fall detection techniques [4, 5, 9], the user needs to wear or carry a device along with them permanently. In such cases, the fall detection method is based on certain readings from these devices where a threshold value is used to determine whether or not the end user has fallen. These approaches are good, but do not consider the physiological parameters of the end users, and are therefore unable to predict a fall due to abnormal health conditions.

Some works tried to use machine learning approaches to detect human falls [10, 12, 13], but their work area is limited to several parameters and is based only on a few chosen algorithms. The works based on fall prediction [14,15] also have genuine concerns. In their two works discussed above, the authors have used the same dataset for their research. However, from our point of view, the dataset is very small for machine learning and data mining techniques. Therefore, it might be possible that, in real life, the trained model will not be able to provide accurate predictions. The authors claimed to get 100% accuracy in their approach with a base of association rules, which is practically not feasible because there is always the possibility of wrong predictions in the most efficient models.

In contrast to these existing works, we try to predict the likelihood that a fall can occur due to abnormal health conditions. Even in case of a strong model, fall detection can only draw medical attention after a fall has occurred. It cannot prevent a fall, whereas precise fall prediction can be a way to prevent these falls and is therefore much more effective than fall detection. If we have an efficient mechanism to know when and for whom a fall is likely to occur, then we can notify the end users in advance and provide certain measures and user guidelines to avoid the fall or at least minimise the risk. This section shows that there was not much work done in the field of fall prevention by prediction. The details of our proposed health parameter-based fall prediction methodology are explained in the next section.

3. Our Approach to Fall Prediction

The aim of the proposed work is to predict the sudden fall of human beings caused by abnormal physiological conditions. In medical terminology, this problem is known as syncope. Before we elaborate on the details of the work done, we briefly discuss the term syncope to understand the effect of abnormal physiological conditions which cause this problem. This section is divided into four subsections: (1) Brief Summary of Syncope, (2). Framework and the Used Dataset, (3) Data Preprocessing, and (4) Proposed Ensemble Algorithm.

3.1. Brief Summary of Syncope. The most frequent reason for syncope is very low blood pressure (hypotension), which prevents the heart from pumping the required amount of oxygen and blood to the brain [16]. Apart from that, a syncope can also occur due to a sudden drop in the heart rate and a lack of adequate blood supply to body parts. However, there can be one or many non-life-threatening factors that may trigger syncope, such as overheating, dehydration and heavy sweating, frequent change in body positions, and exhaustion. Syncope can be observed in any person regardless of their age group. However, it is found that adults over the age of 80 have a higher risk of hospitalization and death [16].

The risk of syncope is higher when people have a known heart disease, brief palpitations (sudden loss of consciousness), frequent fainting, an abnormal cardiac exam, and family history of inheritable conditions. Thus, it is important to study the cause of syncope in terms of a person's health condition. Only then, the probability of a person's falling can be predicted in advance by analysing their health condition in real time. It has been found that the occurrence of syncope is directly related to a person's health conditions. Even stress and pressure have a direct impact on a person's health and might be a reason for syncope. To predict syncope, it is important to understand the health parameters related to it. We also need to know the different ranges or values of these parameters, which lead to increased probabilities of syncope. After a thorough study and discussions with health professionals, we compiled a list of several health parameters and their normal and abnormal ranges of readings. It is important to note that, under normal health readings, there is almost no chance of suffering a syncope. Further research and study also found that syncope also has an indirect relationship with a person's gender. In Table 1, we summarise the individual parameters and the abnormal ranges separately for adult males and adult females.

Table 1: Health parameters related to syncope [17–19].

Parameter	Abnormal range	
	Adult male	Adult female
Heart rate	If not 60–100 bpm	
QRS interval	>0.12 s	
QT interval	>0.40 s	>0.44 s
Systolic BP (top)	<90 mm Hg	
Diastolic BP	<60 mm Hg	
Haematocrit level	If not 42–54%	If not 38–46%
Anemia	If not 12.4–14.9 gm/dL	If not 11.7–13.8 gm/dL
SpO2	If not 95–100%	
Sugar level	If not 70–180 mg/dL	

3.2. Framework and Used Dataset. In this work, we exploit the benefits of machine learning (ML) algorithms. In general, ML algorithms support healthcare-related works through five main applications: detection, diagnosis, prediction, prognosis, and therapy. In our work, we exploit the prediction application of ML algorithms to predict sudden human falls due to unusual health conditions based on medical data. Figure 1 illustrates the proposed work.

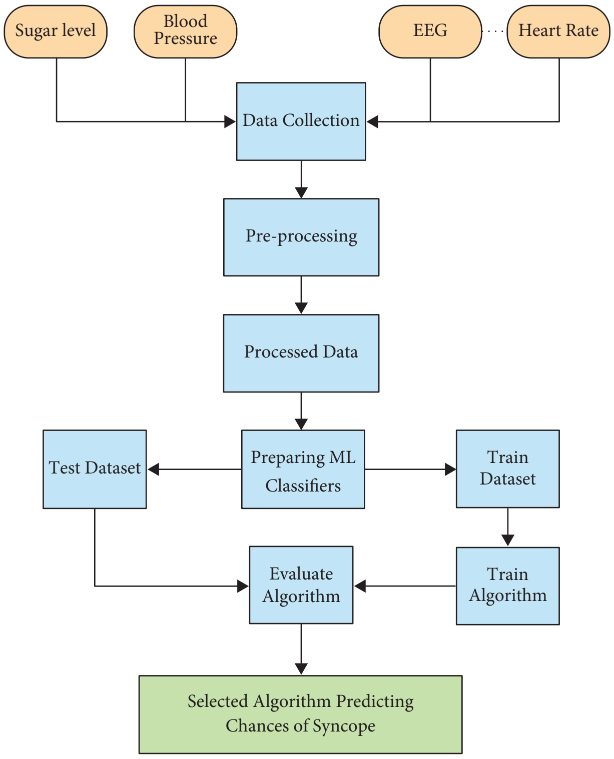

Figure 1: Flow diagram of the proposed methodology.

It has already been discussed that different health parameters, such as heart rate, blood pressure, and sugar level, play an important role in triggering syncope. In our work, we have used a dataset available in *Kaggle*, a Google-owned online community for data scientists founded in April 2010, which allows its users to get and upload datasets. The used dataset is prepared by collecting physiological data of elderly patients from various Chinese Hospitals [20, 21]. However, we cannot directly use this dataset. We first need a data preprocessing phase, where we implement certain methodologies such as outlier detection and normalisation of data to make it compatible for our further processes. Next, we need to identify particularly useful features and attributes; this process is known as feature selection. However, as in our case, the dataset does not contain many attributes; we used all of them. Then, we train various machine learning algorithms using the same processed dataset. Once the models are trained, we evaluate them on several records randomly selected from the dataset and observe their outputs. Based on these outputs, we evaluate all algorithms using certain parameters and select the best one in terms of accuracy of predicting sudden human falls due to abnormal health condition. The original dataset consists of the following attributes: (1) ACTIVITY (activity labels), (2) TIME (monitoring time), (3) SL (sugar level), (4) EEG (EEG monitoring rate), (5) BP (blood pressure), (6) HR (heartbeat rate), and (7) CIRCULATION (blood circulation). The activities considered in the dataset are standing, walking, sitting, falling, cramps, and running. Before employing the learning algorithms, we consider the falling activity as 1 and other activities as 0 in the activity attribute. The implementation of the model has been done using the programming language Python 3.7 with the help of Spyder IDE provided by the Anaconda Environment. The implementation uses Python packages, such as the Numpy, Pandas, and Matplotlib, to support the efficiency of the work. As stated above, the first step of building a machine learning model is to preprocess the data before making it available for the training model. It is believed that using the work that has been used and tested by many users on a large scale makes it the most efficient for our work, saves time, and leaves the least chance for errors. In our work, to implement the machine learning algorithms, we use the Scikit-learn library. It provides a large set of machine learning algorithms for our work.

3.3. Data Preprocessing. There are many outliers in Kaggle's raw dataset. This may affect the efficiency of the model and mislead the results. Therefore, in the data preprocessing phase, we first eliminate those outliers using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. The DBSCAN is a clustering method using unsupervised learning that takes a minimum of two parameters—*eps* and min *_samples*—to divide the data into clusters or groups based on their similarities. They are as follows:

(i) eps: It defines the neighbourhood around a data point; that is, if the distance between two points is lower or equal to "eps," then they are considered as neighbours. If the eps value is chosen too small, then a large part of the data will be considered outliers. If it is chosen very large, then the clusters will merge, and a majority of the data points will be in the same clusters.

(ii) min_samples: It defines the minimum number of neighbours (data points) within the eps radius. The larger the dataset, the larger the value of min_samples that must be chosen. The minimum value of min_samples that must be chosen is at least 3.

In this work, we have tried various combinations of eps and min_samples and appropriate values for eps $=3$ and min_samples $=3$. After outliers are removed, we know that the different parameters are in their respective measurement ranges, which may lead to a dilution in effectiveness. Therefore, we need to normalise these data points and scale them into the same measurement scale for the model to perform accurately.

In order to normalise the dataset, we use the standard linear transformation min-max normalisation technique. In this approach, every value of all the attributes is replaced according to the equation:

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}, \quad (1)$$

where z is the new computed value, x is the original value, $\min(x)$ is the minimum of the selected attribute, and $\max(x)$ is the maximum of the selected attribute. In our work, we used the module "MinMaxScaler" provided by the Scikit-learn library. After outlier detection and normalisation of the dataset, our data is now ready as input to the machine learning algorithms.

3.4. Proposed Ensemble Algorithm. As our previous efforts progressed, we recognised that regression models do not perform well for our problem, as they do not fit properly for complex datasets. They also suffer from underfitting problems and are sensitive to outliers. We realized that our problem requires a classification approach to make better predictions. Thus, we studied the impact of health parameters on the prediction of syncope with the help of classification models in machine learning. We applied the following machine learning algorithms.

3.4.1. Logistic Regression (LR). This is a statistical model for analysing the data with one or more independent variables that help determine an outcome. The goal is to find the best fitting model to establish a relationship between the independent variable(s) and the dependent variable. It uses the sigmoid function, which maps any real-world entity between the range of 0 and 1. One of the significant factors of logistic regression classifier is the threshold value which affects the precision and recall and hence the overall performance (in terms of accuracy) of the algorithm. Generally, we set the threshold value to 0.5 to avoid favouring a particular class label. Mathematically, the sigmoid function is denoted as follows:

$$y = \frac{1}{1 + e^{-x}}, \quad (2)$$

where x is the input (dependent variable), y is the predicted class label, and e is Euler's number (=2.718).

3.4.2. Decision Tree Classifier (DTC). This method can be used for both classification and regression problems. It breaks down the given dataset into smaller subsets and is developed incrementally. In the tress, a decision node and two or more branches and the lead note denote the classification or decision. The method can handle both categorical and numerical data. It tries to mimic human-like thinking based on the tree diagram by selecting the best feature, an Attribute Selection Measure (ASM), to split the dataset's entities. The ASM is a heuristic for selecting the most optimal split that yields the best result with low variance. To stop the splitting of the ever-growing tree, the method of the same class expedition is performed, in which the tree has grown only towards the same class.

3.4.3. Random Forest Classifier (RFC). This is an ensemble learning method that can be used as both classification and regression methods. The random forest algorithm is an enhancement of decision trees that avoids overfitting the dataset by creating random oversampling of data with replacement, aka Bootstrapping, and creates multiple decision trees that are sampled independently. Because of the individuality of the trees, it will reduce the bias and variance with the experience. The support of numerous base learners set this classifier apart from the rest of the classifiers.

3.4.4. K-Nearest Neighbours (KNN). It is a classification algorithm and falls under supervised learning methods. It takes a bunch of labelled points and uses them to learn how to label other points. To label a new point, the labelled points closest to that new point are considered, and these neighbours have to vote, so whichever label most of the neighbours have is the label for the new point (k is the number of considered neighbours). In order to calculate the distance between the data points, KNN uses the Euclidian distance, and the voting is done with the set of (k) minimally reached neighbours only. Researchers use this classifier as it is effortless to implement and yet provides efficient results. Also, it is robust against noisy data and very effective if we have adequate data.

3.4.5. Support Vector Machine (SVM). The method uses a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. One of the significant problems in SVM is identifying the hyperplane on the mapped data. It is generally considered to choose the hyperplane where we have a higher margin between the data points, which reduces the likelihood of misclassification.

3.4.6. Naive Bayes Classifier (NBC). NBC is a classification technique based on Bayes' theorem with an assumption of independence between predictors. If these features depend on each other or the existence of other features, all of these properties independently contribute to the probability. Along with simplicity, NBC is known to outperform even highly sophisticated classification methods.

Cross-validation is used to evaluate the machine learning algorithms based on resampling of data. Only a single value is used as a parameter, k, which defines the number of iterations the resampling needs to perform. In our work, we have used a value of $k = 5$. The general procedure is as follows:

(i) Shuffle the dataset randomly

(ii) Split the dataset into $k = 5$ groups

(iii) For each unique group

Take the group as a test dataset
Take the remaining groups as a training dataset
Fit the model on the training set and evaluate it on the test set
Retain the evaluation score and discard the model

(iv) Summarise the model's performance using the sample of model evaluation scores

Using all the above-mentioned algorithms, we finally proposed an ensemble algorithm—Majority Voting—for syncope prediction. Voting and averaging are the two widely used ensemble algorithms. Averaging is used for regression problems, while voting is used for classification problems such as ours. In the proposed Majority Voting algorithm, the first step is to use a single training dataset and train multiple classification models, as illustrated in Figure 2. Now, every model that has been trained under our ensemble learning process makes a prediction for each instance and the final outcome prediction is the one that receives the largest number of votes from all votes. This method of casting votes from multiple models or algorithms is also known as "Plurality Voting." In this ensemble learning approach, we are allowed to assign some defined weights to all the algorithms to define the importance of every model for the considered application. In our work, we have considered all the weights to be equal. The final outcome of our work is equally based on all algorithms mentioned above. All models are assigned equal weights that are 1. Since the final output should be the prediction value for the chance of syncope, which plays a very important role for all our users, we have decided to predict that the chance of syncope in the final output should be greater than 25%. The pseudocode of the Majority Voting Ensemble method is provided in Algorithm 1. The approach is also denoted as ensemble integration using simple averages in [22] or as mean rule in [23].

4. Results

For the readers' convenience, the result section is divided into three subsections: (1) Description of the Experimental Setup, (2) Comparison of Prediction Efficiency, and (3) Overall Performance Analysis.

4.1. Description of the Experimental Setup. In our work, we have used Python 3 as the programming language with Spyder (ver. 3.3.4) as the working IDE (Integrated Development Environment) provided by Anaconda, which is an open-source distribution for R and Python. The Numpy and Pandas libraries were used for the computational work and work with files such as comma-separated value (CSV) files. For the display of results and plotted graphs, the MatPlot library is used. To keep track of start time and end time and

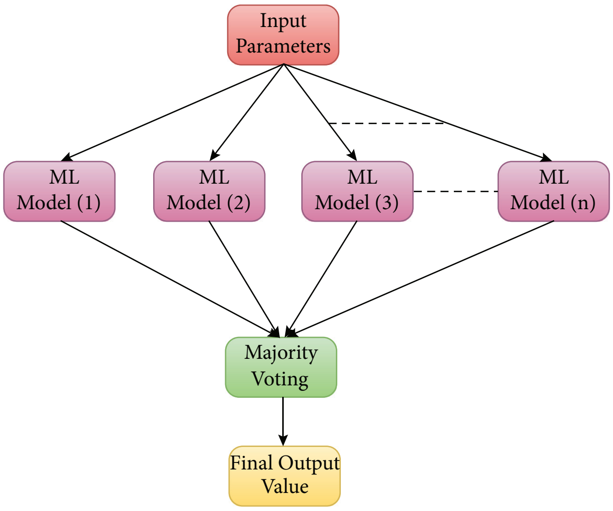

Figure 2: Working of Majority Voting Ensemble algorithm.

```
Output: Predicted Syncope Value
Initialise:
    MLModels = [LR, DTC, RFC, KNN, SVC, NBC];
    AlgoWeights = [1, 1, 1, 1, 1, 1];
    count = 0;
For data in preProcessedDB:
    outputs = [ ];
    For model in MLModels:
        predictedValue = model.predict(data)
        outputs.append(AlgoWeights[count] ∗ predictedValue)
    SyncopeValue = sum(outputs)/count(MLModels) ∗ 100
    Return SyncopeValue
```

Algorithm 1: Majority Voting Ensemble Algorithm.

calculate the execution time for the algorithms, we have used the Time module.

At the beginning of our work, we start the timer whenever an instance of the code is run, and it will be stopped at the end of execution. We have kept all the data related to syncope gathered from Kaggle into a CSV file. In our first step, we randomly shuffle all the data rowwise to avoid any unwanted patterns and provide consistent data for training our models. The next step is to make the data ready for our algorithms. We implement the DBSCAN algorithm as an outlier detection technique using the eps value and min_samples to be 3 for both. We tried with different combinations of eps and min _samples value and found the above values to be the best fit for our work. The DBSCAN algorithm provided us with all the outliers, which were then removed, and all remaining data points were then stored into a separate CSV file as a part of the backup process. The data after removal of outliers consist of 2002 distinct data points of persons' medical health parameters and their syncope attributes. Next, we divide this whole dataset into the dependent and independent variables by attribute. The attributes in columns 1, 2, 3, and 4 were considered independent variables, and the last column, number 5, is kept as the dependent variable. The independent variables are the attributes, such as blood pressure, EEG, and heart rate, while the dependent variable is the attribute which tells whether the person with related medical health parameters has syncope or not, that is, 0 or 1. Before clearly stating that the dataset is processed and ready for machine learning algorithms, it is important to rescale all data; that is, we need to standardise the data on the same scale. We scale all the dependent variables, or let us say the medical health parameter readings, in the open bracket of [0, 1]. For this purpose, we use Scikit-learn library's MinMaxScaler module.

In the next step, we divide our preprocessed data into two groups: the training data and the test data. The training data is used to train the models; the models use this data to find patterns and relationships within the data. Once the training process is completed with the training data, we test the efficiency of the machine learning model using the test data. For our work, we have observed that the models work best when we divide the dataset into training and test datasets in the ratio 3 : 1, respectively. Dividing the preprocessed dataset in the above ratio gives 1501 data points or records for the training dataset and the remaining 501 data points for the testing of models. We also propose an ensemble learning approach, where our final output or prediction is derived from the results of the considered six machine learning algorithms.

4.2. Comparison of Prediction Efficiency. Figures 3–8 compare the actual and predicted results in case of all six machine learning algorithms. In each of these figures, the x-axis shows the instances of test data points ranging from 0 to 500 in our case. The y-axis shows the chance of fall or syncope value which is either 0 or 1. In the middle of each figure, we provide the confusion matrix, which is a matrix containing information about the performance of a classification algorithm on a set of test data for which the true values are already known. A confusion matrix consists of four values as follows:

(1) True positive (TP) : predicted as positive and was actually positive

(2) False negative (FN) : predicted as negative but was actually positive

(3) True negative (TN) : predicted as negative and was actually negative

(4) False positive (FP) : predicted as positive but was actually negative

From Figures 3–8, it is evident that the TN value of LR algorithms is 378, which is the highest value among all the algorithms considered in this study. Despite this fact, the LR algorithm is not a good choice because it also has the second-highest FN value. A high FN value means a wrong prediction, which is not at all desirable in any case. Furthermore, the TP value is the lowest for LR. Therefore, LR is not a good choice at all for predicting syncope. The same holds for the NBC algorithm. The performance of SVM is better than LR and NBC in terms of all matrix values. However, the DTC, RFC, and KNN algorithms provide really good results. Among these three, KNN obtains the best results in terms of TP value. The output for the Majority Voting Ensemble algorithm and its confusion matrix are depicted in Figures 9 and 10, respectively. We can see that it also performs similarly well to DTC, RFC, and KNN.

Using the Majority Voting algorithm, the final output for the dependent variable is calculated for multiple individual machine learning algorithms, which are then used to determine the final value for the chances of syncope. In this approach, we count the number of votes from the algorithms which predicted that the user is having a syncope and then used this count to calculate the final value by using the below equation:

$$\text{chance of syncope} = \left(\frac{\text{count votes}}{6}\right) \cdot 100. \quad (3)$$

To better understand how the system works, let us suppose that there is a person X, who is interested to know their chances of having a syncope. We gather all required health parameters. When all the independent machine learning algorithms are executed on person X's health parameters, they predict certain binary outputs in the form of 0 or 1. Since we have used 6 machine learning algorithms, we now have 6 binary outputs. Let us assume the outputs are {0, 1, 1, 0, 1, 1}. Our model will now count the number of votes in favour of 1, that is, the person having a syncope, which is 4 in case of person X and this count is passed into the above formula to calculate the chances of syncope for person X. Hence, the chances of syncope are $(4/6)^*$ 100, which is 66.66%. If we come up with the conclusion that the predicted chances of syncope are greater than or equal to 25%, then we assume that the person is having a severe risk of syncope. This person must be taken care of and also should take preventive measures to avoid and prevent the damage caused by the occurrence of syncope.

In our experiments, we have taken the value of $K = 5$ and for all the 5 iterations the recorded result, that is, the percentage accuracy, is depicted in Figure 11, where we can see that for every machine learning model used in our work, there are 5 percentage accuracy scores. Since we used $K = 5$ in our work, we obtained 5 percentage accuracy scores upon each iteration, that is, from iteration 0 to iteration 4. From Figure 11, it is clearly visible that there are no predetermined patterns in the dataset and all the iterations yield approximately similar results.

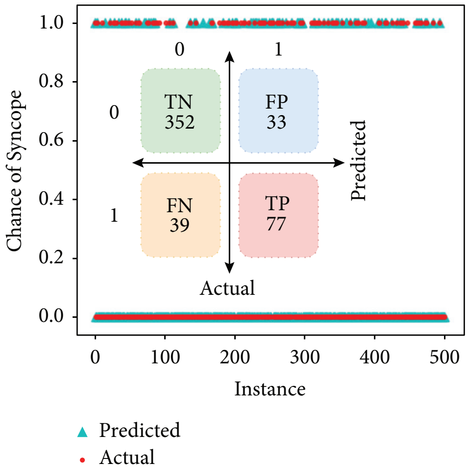

FIGURE 3: Performance of KNN.

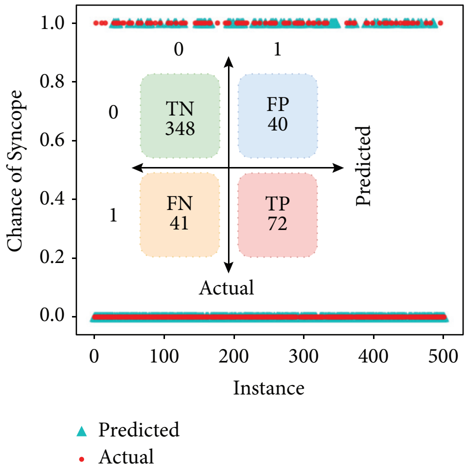

FIGURE 4: Performance of DTC.

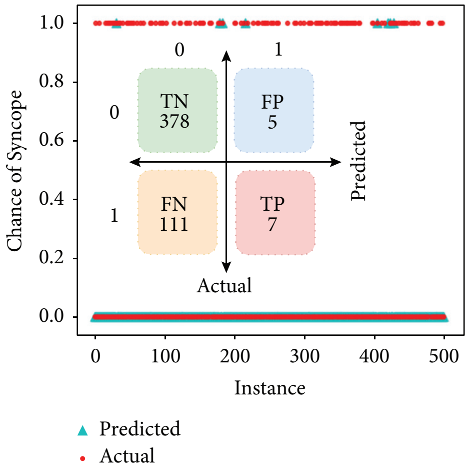

Figure 5: Performance of LR.

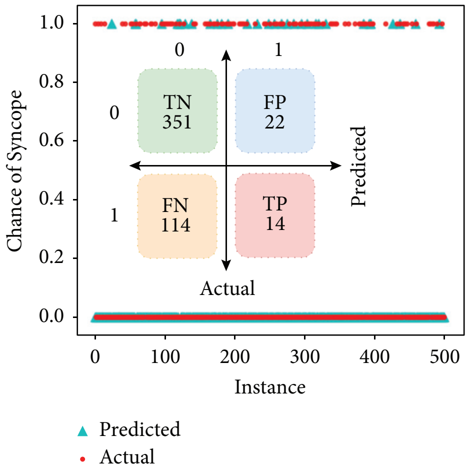

Figure 6: Performance of NBC.

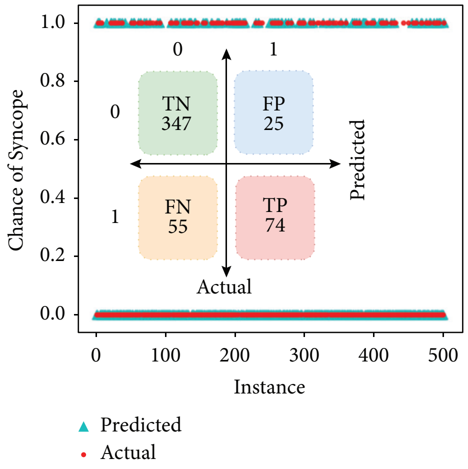

Figure 7: Performance of RFC.

4.3. *Overall Performance Analysis.* The specification of the system in which we performed all the simulations are summarised as follows:

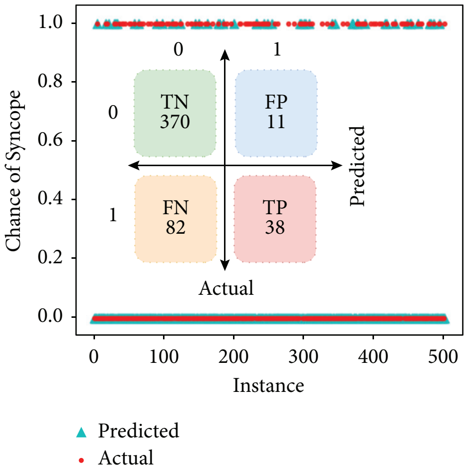

Figure 8: Performance of SVM.

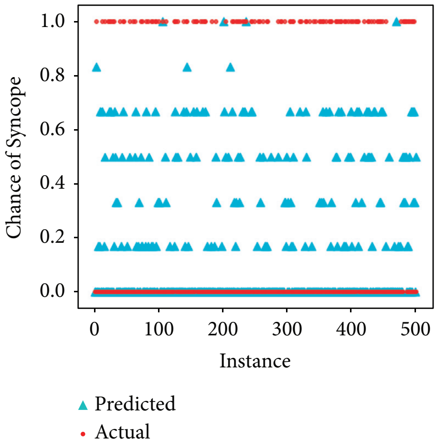

Figure 9: Performance of Majority Voting.

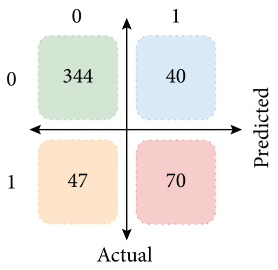

Figure 10: Confusion matrix of Majority Voting.

(i) System specification: Apple MacBook Pro, Intel Core i5, 16 GB RAM, 512 HDD

(ii) Operating System: macOS Catalina ver. 10.15.3

Based on our above observations and the Python simulation, we calculated various metrics such as precision score, recall value, percentage accuracy, and F1-score. The formulas to find the values of these metrics are as follows:

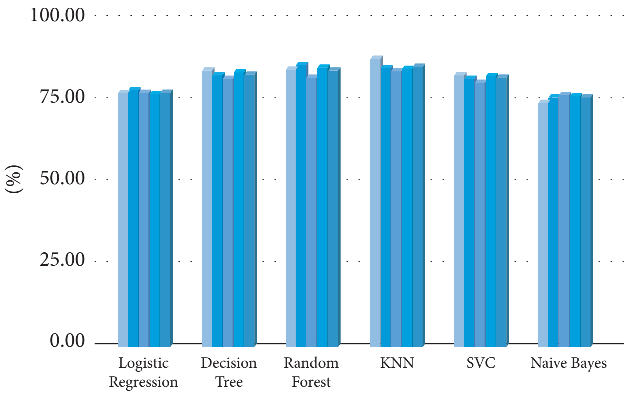

Figure 11: Results of 5-fold cross-validation.

Table 2: Performance scores for all algorithms.

Model	Precision	Recall	$F1$-score	Accuracy (%)
Logistic regression	0.583	0.059	0.107	76.84
Decision tree classifier	0.642	0.637	0.64	83.83
Random forest classifier	0.747	0.573	0.649	84.03
K-Nearest Neighbour (KNN)	0.7	0.663	0.681	85.62
Support vector machines (SVM)	0.775	0.316	0.449	81.43
Naïve Bayes classifier	0.388	0.109	0.170	72.85
Majority voting	0.722	0.702	0.712	87.42

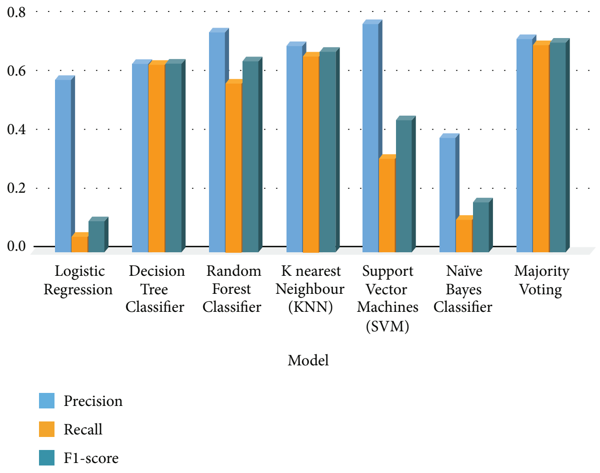

Figure 12: Comparison of precision, recall, and F1-score.

$$
\begin{aligned}
\text{precision} &= \frac{\text{true positive}}{\text{true positive} + \text{false positive}},\\
\text{recall} &= \frac{\text{true positive}}{\text{true positive} + \text{false negative}},\\
F1 - \text{s core} &= 2\frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}},\\
\text{accuracy} &= \frac{\text{true positive} + \text{true negative}}{\text{total}}.
\end{aligned}
\tag{4}
$$

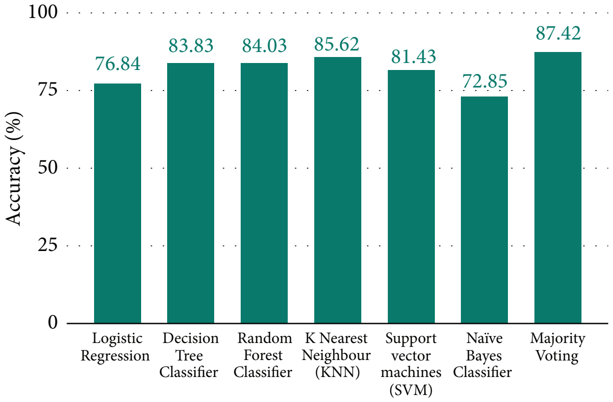

Figure 13: Comparison of percentage accuracy.

Based on the above-discussed formulas, the performance scores for all machine learning algorithms are shown in Table 2. In order to get a clear picture of the above data, we plotted the values of all the algorithms, including the ensemble learning Majority Voting system, in the form of a bar graph in Figure 12. The accuracy comparison is illustrated in Figure 13.

Although the comparison shows the best results for the Majority Voting classifier, the differences in accuracy do not appear big enough to show superiority with statistical significance of this classifier over the other classifiers considering k-cross-validation with the rather small sample of 2002 data points in total.

The specific advantages of the proposed system are as follows:

(i) It reduces the likelihood of model overfitting by looking at the result considering multiple classifiers and helps to achieve optimal accuracy.

(ii) The proposed model increases model accuracy and performance because of the multiple support of base learners (used classifiers). The likelihood of real-world prediction performance also increases as it has already learned from its experience through multiple base models.

(iii) The base algorithms used to differ in their internal computation and hence reduce the probability of overall bias and variance.

5. Conclusion

In this paper, we studied the effect of various machine learning algorithms in predicting syncope, which is a situation of temporary loss of consciousness caused by abnormal changes in health parameters and a major reason for sudden accidental falls. The proposed Majority Voting Ensemble scheme achieved an accuracy of 87.42% in predicting syncope, which is more than the accuracy of single prediction algorithms such as KNN, DTC, LR, NBC, RFC, and SVM. Although we believe that these results are already rather good and promising for practical application, further improvements are possible by using a wider range of data and exploring a broader range of machine learning techniques. In the future, we aim to incorporate more health parameters in the proposed model to make it more accurate. Further important parameters such as height, weight, and sex can also be considered in the future. With further advancement of sensor-based real-time health monitoring systems and cloud computing, it will also be possible to develop an end-to-end working prototype which will be able to alert end users in real time when predicting the likelihood of a sudden accidental fall due to health abnormalities.

Data Availability

The dataset used in this paper for machine learning can be assessed at Kaggle: https://www.kaggle.com/pitasr/falldata.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] World Health Organization, *WHO Global Report on Falls Prevention in Older Age*, World Health Organization, Geneva, Switzerland, 2008.

[2] United Nations, *World Population Prospects: The 2017 Revision, Key Findings and Advance Tables*, Department of Economics and Social Affairs PD: United Nations, New York, NY, USA, 2017.

[3] United Nations, *World Population Ageing 2019: Highlights*, Department of Economics and Social Affairs, Population Division, United Nations, New York, NY, USA, 2019.

[4] F. Wu, H. Zhao, Y. Zhao, and H. Zhong, "Development of a wearable-sensor-based fall detection system," *Hindawi International Journal of Telemedicine and Applications*, vol. 2015, Article ID 576364, 11 pages, 2015.

[5] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou, "Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information," in *Proceedings of the IEEE 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, pp. 138–143, Berkeley, CA, USA, June 2009.

[6] M. Alwan, P. J. Rajendran, S. Kell et al., "A smart and passive floor-vibration based fall detector for elderly," in *Proceedings of the IEEE 2006 2nd International Conference on Information & Communication Technologies*, pp. 1003–1007, Damascus, Syria, April 2006.

[7] A. Sixsmith and N. Johnson, "A smart sensor to detect the falls of the elderly," *IEEE Pervasive computing*, vol. 3, no. 2, pp. 42–47, 2004.

[8] A. Sixsmith, N. Johnson, and R. Whatmore, "Pyroelectric IR sensor arrays for fall detection in the older population," in *Journal de Physique IV*vol. 128, , pp. 153–160, EDP sciences, 2005.

[9] S. Moulik and S. Majumdar, "FallSense: an automatic fall detection and alarm generation system in IoT-enabled environment," *IEEE Sensors Journal*, vol. 19, no. 19, pp. 8452–8459, 2018.

[10] M. Waheed, H. Afzal, and K. Mehmood, "NT-FDS-A noise tolerant fall detection system using deep learning on wearable devices," *Sensors*, vol. 21, no. 6, p. 2006, 2021, https://www.mdpi.com/1424-8220/21/6/2006.

[11] N. Zurbuchen, A. Wilde, and P. Bruegger, "A machine learning multi-class approach for fall detection systems based on wearable sensors with a study on sampling rates selection," *Sensors*, vol. 21, no. 3, p. 938, 2021, https://www.mdpi.com/1424-8220/21/3/938.

[12] A. Iazzi, M. Rziza, and R. Oulad Haj Thami, "Fall detection system-based posture-recognition for indoor environments," *Journal of Imaging*, vol. 7, no. 3, p. 42, 2021, https://www.mdpi.com/2313-433X/7/3/42.

[13] A. Collado-Villaverde, M. Cobos, P. Muñoz, and D. F. Barrero, "A simulator to support machine learning-based wearable fall detection systems," *Electronics*, vol. 9, no. 11, p. 1831, 2020, https://www.mdpi.com/2079-9292/9/11/1831.

[14] M. Guftar, S. H. Ali, A. A. Raja, and U. Qamar, "A novel framework for classification of syncope disease using K-means clustering algorithm," in *Proceedings of the IEEE 2015 SAI Intelligent Systems Conference (IntelliSys)*, pp. 127–132, London, UK, November 2015.

[15] M. Guftar and U. Qamar, "A rule based expert system for syncope prediction," in *Proceedings of the IEEE 2015 SAI Intelligent Systems Conference (IntelliSys)*, pp. 559–564, London, UK, November 2015.

[16] American Heart Association, *Symptoms, Diagnosis and Monitoring of Arrhythmia*, American Heart Association, Inc., Chicago IL, USA, 2014, https://www.heart.org/en/health-topics/arrhythmia/symptoms-diagnosis–monitoring-of-arrhythmia.

[17] T. Al-Aama, "Falls in the elderly: spectrum and prevention," *Canadian Family Physician*, vol. 57, no. 7, pp. 771–776, 2011.

[18] L. G. Robert, “Evaluation of syncope,” *American Family Physician*, vol. 84, 2011, https://www.aafp.org/afp/2011/0915/afp20110915p640.pdf.

[19] R. T. Saccilotto, C. H. Nickel, H. C. Bucher, E. W. Steyerberg, R. Bingisser, and M. T. Koller, “San Francisco Syncope Rule to predict short-term serious outcomes: a systematic review,” *Canadian Medical Association Journal*, vol. 183, no. 15, pp. E1116–E1126, 2011.

[20] Kaggle, *Machine Learning Datasets: Fall Detection Data from China*, Kaggle, San Francisco, CA, USA, 2017, https://www.kaggle.com/pitasr/falldata.

[21] A. Özdemir and B. Barshan, “Detecting falls with wearable sensors using machine learning techniques,” *Sensors*, vol. 14, no. 6, pp. 10691–10708, 2014.

[22] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. D. Sousa, “Ensemble approaches for regression,” *ACM Computing Surveys*, vol. 45, no. 1, pp. 1–40, 2012.

[23] R. Polikar, “Ensemble learning,” in *Ensemble Machine Learning*, pp. 1–34, Springer, Boston, MA, USA, 2012.