


Research Article

Novel Stacking Classification and Prediction Algorithm Based Ambient Assisted Living for Elderly

Jinesh Padikkapparambil ¹, Cornelius Ncube ², Firoz Khan ³,
Lakshmana Kumar Ramasamy ⁴ and Yomiyu Reta Gashu ⁵

¹Higher Colleges of Technology, UAE

²British University in Dubai, UAE

³Higher Colleges of Technology, British University in Dubai, UAE

⁴Hindusthan College of Engineering and Technology, Coimbatore, India

⁵Tepi University, Addis Ababa, Ethiopia

Correspondence should be addressed to Yomiyu Reta Gashu; yomiyu@mtu.edu.et

Received 31 March 2022; Revised 23 April 2022; Accepted 23 May 2022; Published 18 June 2022

Academic Editor: Kuruva Lakshmana

Copyright © 2022 Jinesh Padikkapparambil et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recognition of human activity is a significant area of research with numerous uses. In developed countries, the rising age of citizens requires the improvement of the medical service structure, which raises the price of resources, both financial and human. In that sense, ambient assisted living (AAL) is a relatively novel information and communication technology (ICT) that presents services and recognizes various products that enable older people and the disabled to live autonomously and improve their quality of life. It further assists in reducing the cost of hospital services. In the AAL environment, various sensors and devices are fixed to gather a broad range of data. Moreover, AAL will be the motivating technology for the latest care models by acting as an adjunct. This will become thought-provoking research in a fast-growing world, but exploring different ADL and self-classification will become a major challenge. This paper proposed a Novel Stacking Classification and Prediction (NSCP) algorithm based AAL for the elderly with Multi-strategy Combination based Feature Selection (MCFS) and Novel Clustering Aggregation (NCA) algorithms. This paper's main aim is to recognize the activity of older people, such as standing, walking, sitting, falling, cramps, and running. The dataset is derived from the Kaggle repository, which refers to data collection from wearable IoT devices. The experimental outcomes demonstrate that the MCFS, NCA, and NSCP algorithms work more efficiently than existing feature selection, clustering, and classification algorithms, respectively, regarding the accuracy, sensitivity, specificity, precision, recall, F-measure, and execution time dataset size and the number of features. Furthermore, the NSCP algorithm provided high accuracy, precision, recall, and F-measure are 98%, 0.96, 0.95, and 0.98, respectively.

1. Introduction

The ageing population is one of the world's key concerns because of its leading socioeconomic impacts [1]. Over ageing can guide numerous issues ranging from fundamental practical disabilities to serious health issues, for example, arthritis, diabetes, and depression. Also, medical issues and dependence on family members and caregivers for their everyday activities can lead to shame and inadequate nutrition [2]. As a result, the need for nursing homes has been

rising over the past decade for continuing care and continued evaluation of physical and mental health [3].

Although this does not entirely solve their issues, technology is a tool that could give them an autonomous and happy life while simultaneously providing precise and well-timed personal care by medical home staff [4]. There have been rapid growth in ambient assisted living (AAL) technologies to address some of the issues mentioned [5]. AAL is a growing tendency to use novel services, securities, and products that enable a higher quality of independent

living [6]. AAL exemplifies hospitalized issues because of an ageing population. Patient monitoring provides an independent life.

It further assists in reducing the cost of hospital services to boost the standard of living of the elderly [7]. In the AAL surroundings, various sensors are embedded to gather a broad range of data [8]. It is not easy to diagnose and develop the activities of daily living (ADL) database in the AAL surroundings. At first, we must clearly understand what functions the user is doing, how they function, and their progression. Monitoring ADL is to differentiate any medical problems from those with minimal exercise. AAL's main objective is to expand the life expectancy of seniors in their favored surroundings using personal health monitoring systems utilizing information and communication technologies (ICTs) [9, 10].

In addition to looking at how care is provided, AAL also stimulates research into more versatile living conditions, which are becoming more innovative manners of ageing. In addition, AAL will serve as a complementary technology to inspire the latest maintenance models. It can become a thought-provoking study; however, searching various ADL and self-categorization can become the main challenge. This paper proposed a Novel Stacking Classification and Prediction (NSCP) algorithm based AAL for the elderly with Multi-strategy Combination based Feature Selection (MCFS) and Novel Clustering Aggregation (NCA) algorithms. In a real-world dataset, the representation of data constantly uses many features. However, some of them may be related to the goal idea.

Moreover, some features may be interrelated redundancies; therefore, modeling does not necessarily cover them all; by being interdependent, if some of them are connected separately, many of the features reveal significant data precision. Feature selection is a method of detecting and removing unwanted and inappropriate data [11]. It reduces the dimension of the dataset and allows learning techniques to operate more efficiently and quickly [12].

In some cases, the accuracy of future classification may increase. Important goals of feature selection [13] are as follows: (1) improving forecast accuracy, (2) eliminating unnecessary features, and (3) minimizing the use of time during analysis. Figure 1 illustrates the feature selection.

The clustering technique clusters the data instances into subsets to cluster similar instances together, while different instances belong to other groups [14]. For example, let us know the clustering technique of the real-world instance of the supermarket [15]: When we visit any supermarket, we can view that the items with related usage are grouped. For example, the groceries are grouped in one section, and personal care items are in other sections. Likewise, laundry/detergents, insect repellents, scrubbers, etc., are grouped in separate sections in house care sections. Consequently, we could effortlessly determine things. The clustering technique further works similarly [16]. Other instances of clustering are grouping documents along with the topic. Figure 2 shows the clustering.

Classification categorizes specific data; it could be done on structured and unstructured data [17]. The procedure

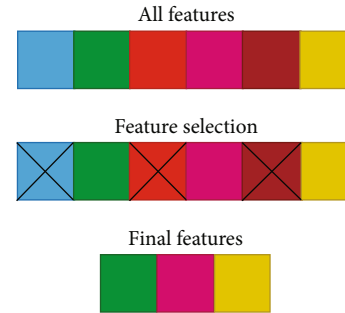


FIGURE 1: Feature selection.

begins with calculating the class of data instances. Classes are frequently known as a label. Classification predictive modeling predicts the output label from the input data [18]. The primary aim is to determine which class the novel data belongs to. We will try to comprehend this with the example shown in Figure 3.

Diagnosis of heart disease could be discovered like a classification issue, a binary classification, because there could only be two classes: heart disease or no heart disease [19]. The classification algorithm requires training data to comprehend how the input data are associated with the class. Moreover, the classification algorithm is trained precisely, and it could be utilized to diagnose whether a particular patient has heart disease or not.

The MCFS algorithm combines five strategies for feature selection, namely, information gain (IG), Fisher score, min-max normalization (MMN), correlation coefficient (CC), and mean absolute deviation (MAD). The NCA algorithm aggregates three clustering algorithms: K-means, expectation-maximization (EM), and density-based spatial clustering of applications with noise (DBSCAN). Furthermore, the NSCP algorithm used a stacking classifier, which contained the repeated incremental pruning to produce error reduction (RIPPER), multinomial logistic regression (MLR), $Dl4jMlpClassifier$ as a base classifier, and Naïve Bayes classifier as meta-classifier.

The remaining sections are explained as follows. The previous works related to fall detection data with feature selection, clustering, and classification are discussed in Section 2. Section 3 discusses the problem definition. Section 4 addresses the description of the dataset taken for this paper. Section 5 discusses the Novel Stacking Classification and Prediction (NSCP) algorithm based AAL for the elderly with Multi-strategy Combination based Feature Selection (MCFS) and Novel Clustering Aggregation (NCA) algorithms. Section 6 presents the simulation results of the proposed work. Lastly, Section 7 provides the conclusion.

2. Related Works

This section reviews last fall detection data or human activity recognition with feature selection, clustering, and classification techniques.

2.1. Feature Selection. Liu et al. [20] offered a simple activity feature selection technique using the Pearson correlation

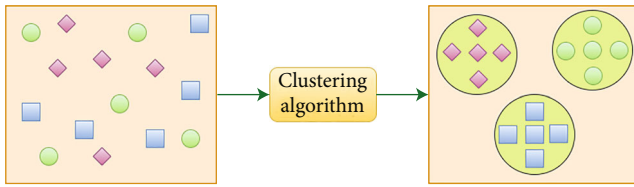


FIGURE 2: Clustering.

coefficient (PCC). First, the daily activity feature is sighted like a vector using the PCC formula. Second, the degree of correlation among everyday activity features is attained with the PCC formula. Finally, the unwanted features are eliminated through the relation among the everyday activity features. Two separate datasets are accepted, alleviating the dataset connection and sensor configuration utilized. Three ML algorithms are used to assess the technique's effectiveness in activity recognition. Experimental results demonstrated that this technique gives higher recognition ratios and reached an average of 1.56% and 2.7% F-measures.

Helmi et al. [21] presented a proficient HAR system utilizing the lightweight feature selection technique to improve the HAR classification procedure. An advanced feature selection system known as GBOGWO targets enhancing the effectiveness of the gradient-based optimizer (GBO) technique using gray wolf optimizer (GWO) operators. Initially, GBOGWO is utilized to choose the convenient features; after that, the support vector machine (SVM) is utilized to categorize the functions. Extensive tests were performed utilizing the famous WISDM and UCI-HAR datasets to evaluate the effectiveness of GBOGWO. Overall, the results demonstrated that GBOGWO improved the classification precision with an average precision of 98.13%.

Nguyen et al. [22] provided a new technique for recognizing functions using sensor placement feature selection. This technique is implemented to solve the multisensor integration data of wearable sensors positioned at various human body spots. Accurately, the technique could extract the most excellent features that exemplify each activity related to the body sensor position to identify everyday life activities. They initially preprocess the dataset using a low-pass filter. After taking out different features, feature selection techniques are used individually in each sensor's feature set to get the finest feature set for each body position. After that, they explore the relevance of the features in each package to improve the set of features. Lastly, it categorizes the thirteen functions into an optimal feature set, consisting of four body positions. The test results get an overall precision of 95.6% using this technique in the benchmark dataset. Furthermore, the outcomes demonstrated that selecting the feature for each sensor location could decrease the calculation time for the feature test step; a higher accuracy ratio can also be attained.

2.2. Clustering. Manzi et al. [23] provided an activity recognition framework using skeleton data from an in-depth camera. The framework used machine learning algorithms to categorize the actions explained by a collection of some essential poses. The training phase generates multiple tech-

niques associated with clustered postures using the multi-class SVM trained by sequential minimum optimization (SMO). The classification stage uses the X-means technique to diagnose the finest number of clusters. This paper's primary objective is to execute activity recognition utilizing features using a limited number of data levels isolated from each activity event; second, it goals to estimate the small number of frames required for good classification. The scheme was assessed on two open-source datasets, the Telecommunication Systems Team (TST) fall detection data and the Cornell Activity Dataset (CAD-60). The number of clusters required to model all instances ranges from 2 to 4 elements. This technique provides 93.3% accuracy.

Cruciani et al. [24] combined the semi-population-based technique with user adaptation to provide a customization technique. Customization is attained by the following. First, this technique discovers a subset of users from the obtainable population like the finest candidates for launching the classification algorithm to the target user. Then, a Sami population neural network classification algorithm is trained to utilize data from this subgroup of users. The network weights of the classifier are then updated utilizing a minimum number of labelled data from the target user, who enables customization. This technique was verified in publicly available big data gathered in an independent living environment. The personalized technique improved the total F-measure to 74.4% compared to 70.9% when utilizing the general not customized technique.

Fáñez et al. [25] presented a fall detection technique; an effortless finite level machine is utilized to procedure acceleration data in sliding windows and extract features from this data when a fall-like event is detected. Utilizing the K-means clustering and SVM and KNN classifiers, the event is categorized as fall or not fall. This study assessed the effectiveness of these clustering and classification techniques. It utilizes a novel dataset, data collected through a wrist-worn device and utilized through many members of the investigating group. This fall detection technique attained 87.50% accuracy.

2.3. Classification. Li et al. [26] approached the performance range by using individual sensors, particularly for categorizing related functions, by activating a data combination of features extracted from experimental data gathered through various sensors, i.e., micro-Doppler radar, a tri-axial accelerometer, and a deep camera. First-round outcomes confirmed that connecting data from multiple sensors enhanced the total effectiveness of the technique. The accuracy achieved through this fusion technique increases by 11.2 percent compared to radar-only applications and by 16.9 percent compared to the accelerometer. Moreover, adding features from the RGB-D kinect sensor improves the total classification accuracy to 91.3%.

Celli et al. [27] utilized four techniques to classify human functions. These techniques are the artificial neural network (ANN), KNN, the quadratic SVM (QSVM), and the ensemble bagged tree (EBT). Novel features that enhance the classification technique's effectiveness are taken out from the energy spectral density of the accelerator. Acceleration data is merely utilized for activity recognition. Their outcomes

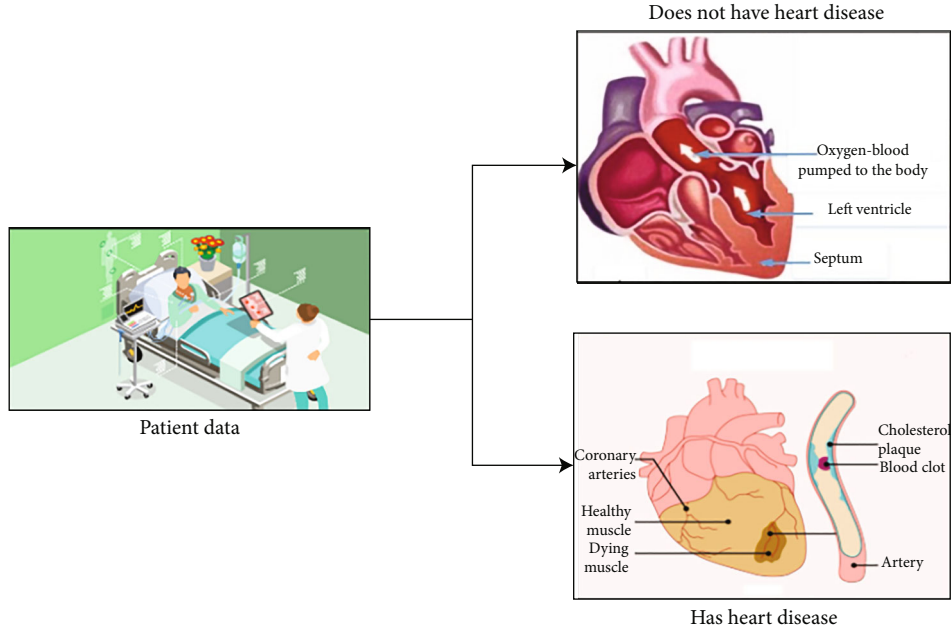


FIGURE 3: Classification.

TABLE 1: Related work summary.

Sr.no.	Author	Year	Findings
1	Liu et al. [20]	2020	The Pearson correlation coefficient technique was utilized for feature selection. It gives higher recognition ratios and reached an average of 1.56% and 2.7% F-measures
2	Helmi et al. [21]	2021	Hybrid GBO and GWO techniques were utilized for feature selection. As a result, it attained 98.13% precision
3	Nguyen et al. [22]	2018	Position-based feature selection algorithm was used. It attained 95.6% precision
4	Manzi et al. [23]	2017	Dynamic clustering technique used for skeleton data. It attained 93.3% accuracy
5	Cruciani et al. [24]	2020	A cluster-based semipopulation technique was proposed. It attained a 74.4% F-score
6	Fáñez et al. [25]	2019	K-means clustering, SVM, and KNN algorithms were used. It attained 87.50% accuracy
7	Li et al. [26]	2017	Multi-sensor data fusion technique is utilized. It attained 91.3% accuracy
8	Chelli et al. [27]	2019	KNN, ANN, QSVM and EBT techniques were utilized. It attained 81.2%, 87.8%, 93.2%, and 94.1%, respectively
9	Yacchirema et al. [28]	2019	Decision trees, ensemble, logistic regression, and Deepnets are used. It attained 94% accuracy

showed that the KNN, ANN, QSVM, and EBT techniques achieved 81.2%, 87.8%, 93.2%, and 94.1%, respectively.

In the IoT-Fall system proposed by Yacchirema et al. [28], we used a 3D-printed accelerometer embedded in the 6LowPAN wearable device, proficient in taking real-time data of ageing volunteers actions. Four machine learning techniques used to present high performance in fall detection, namely, decision tree, ensemble, logistic regression, and deep net, are assessed based on AUC, ROC, train time, and test time. In addition, acceleration measurements are developed and examined at the network edge utilizing an ensemble-based forecast technique discovered as the appropriate forecast for fall detection. Test outcomes for compilation data, executable services, data study and alert, emergency service, and cloud services demonstrated that their framework attained 94% accuracy.

Furthermore, Table 1 demonstrates the outline of the related work.

3. Problem Definition

Among machine learning issues, big data, particularly numerous features, is rising these days. Numerous researchers focus on experiments to solve these issues to extract essential features from these big data. Statistical techniques were utilized to reduce noise and unwanted data. However, we do not utilize the features to train the algorithm. We can upgrade our algorithm with features relevant and unwanted; thus, feature selection plays an essential role. Feature selection is the method of identifying and deleting irrelevant data. It decreases the dimensionality of the information and might permit learning techniques to function quicker and more efficiently. Thus, a proficient feature selection technique is required.

To recognize the activity of older people, a proficient and precise classification and prediction algorithm are essential. Before classification, a clustering technique is needed to

TABLE 2: Dataset feature description.

Feature	Description	Range
Time	Monitoring time	1954.23 to 35031.2
SL	Sugar level	47.713 to 1005950
EEG	EEG monitoring rate	-23313 to 16900
BP	Blood pressure	0 to 533
HR	Heart beat rate	33 to 782
Circulation	Blood circulation	5 to 30042
Activity	Activity classification	0, 1, 2, 3, 4, 5 (0 standing, 1 walking, 2 sitting, 3 falling, 4 cramps, and 5 running)

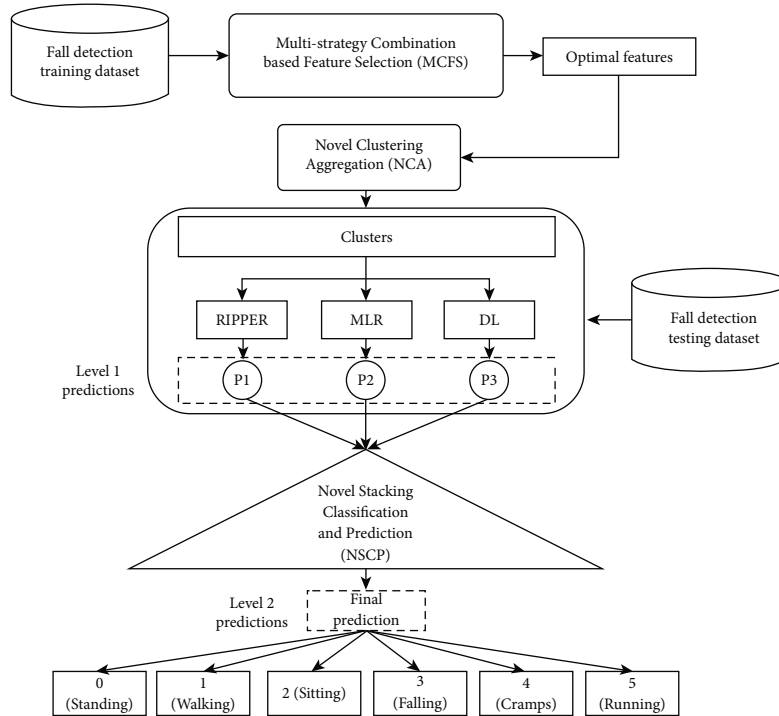


FIGURE 4: Flow diagram of the proposed work.

enhance the classification's effectiveness, precision, and performance. The clustering technique clusters the data instances into subsets to cluster related instances jointly, while various records belong to various sets. Unfortunately, the availability of an enormous compilation of clustering techniques from the literature could confuse specialists attempting to pick a proper algorithm for a dataset. Additionally, no clustering algorithm could generally solve all issues, for example, cluster shape, noise, or density. To deal with these issues, an efficient clustering aggregation algorithm is needed.

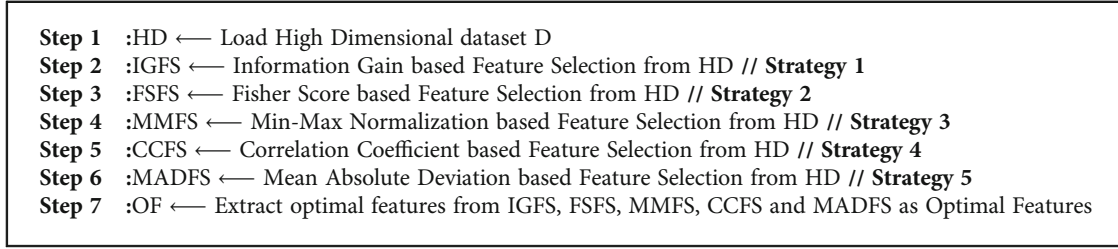
4. Dataset Description

The "Fall Detection Data from China" is utilized in this section and attained from the Kaggle machine learning repository [29]. This paper deals with the problem of categorizing various activities as a fraction of a scheme developed to meet the requirement for a wearable device to gather data for fall and near-fall investigation. Consequently, four fall trajectories (forward, backwards, left, and right), three normal activ-

ities (standing, walking, and lying down), and circumstances near the fall were detected. Falls are a serious public health issue and can be life-menacing. Therefore, this paper executes an automatic fall detection scheme with wearable motion sensor units raised on the subject's body at six stages. Each unit includes three tri-axial devices (compass or magnetometer, gyroscope, and accelerometer). Fourteen volunteers performed standardized activities containing 20 voluntary falls and 16 activities of daily life (ADLs), resulting in an extensive database of 2520 trials. Furthermore, the dataset has seven features: monitoring time, sugar level, EEG monitoring rate, blood pressure, heartbeat rate, blood circulation, and activity classification, as shown in Table 2.

5. Methodology

This section proposes a Novel Stacking Classification and Prediction (NSCP) algorithm based AAL for the elderly with Multi-strategy Combination based Feature Selection (MCFS) and Novel Clustering Aggregation (NCA) algorithms. The flow diagram is shown in Figure 4.



ALGORITHM 1: Multi-strategy Combination based Feature Selection (MCFS)

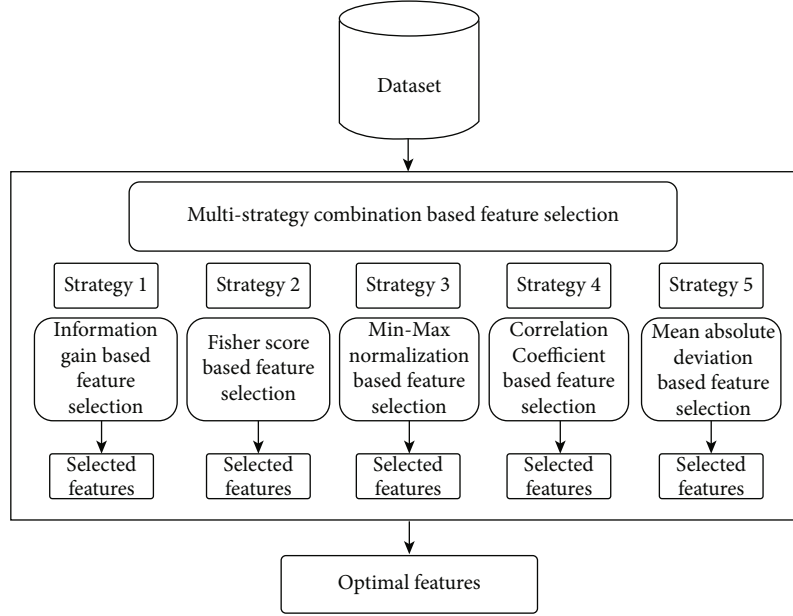


FIGURE 5: Flow diagram of the proposed work.

5.1. Multi-strategy Combination-Based Feature Selection (MCFS). This section presents Multi-strategy Combination based Feature Selection (MCFS) to reduce the dataset's dimensionality. Given a high dimensional dataset $F = \{F1, \dots, FD\}$ containing D features, $FS1 = \{F11, \dots, F1D\}$ is the feature sequence ranked by information gain, $FS2 = \{F21, \dots, F2D\}$ is the one by Fisher score, $FS3 = \{F31, \dots, F3D\}$ is the feature sequence ranked by min-max normalization, $FS4 = \{F41, \dots, F4D\}$ is the feature sequence ranked by correlation coefficient, and $FS5 = \{F51, \dots, F5D\}$ is the feature sequence ranked by mean absolute deviation. Thus, we use the union method on the lowest $C\%$ of the two sequences and extract them from the unique feature sets to filter low-scored features provided by least both feature sequences. The novel feature subset FS after merged feature selection could be described as follows: $FS = F - \{C\%\{FS1\} \cup C\%\{FS2\} \cup C\%\{FS3\} \cup C\%\{FS4\} \cup C\%\{FS5\}\}$. Based on these selected best features, the dimensionality of F is reduced. Algorithm 1 shows the proposed Multi-strategy Combination based Feature Selection.

Figure 5 demonstrates the flow diagram of the MCFS algorithm.

5.2. Information Gain-Based Feature Selection. Information gain is one of the essential strategies that can be utilized

for feature selection by assessing each variable's gain in the target variable's situation [30]. Information gain utilizes entropy to discover the split point and the feature to split, as shown in Equation (1). Entropy is the nonexistence of order or the ability to be predicted. It is the measurement of impurity of a group of instances. Furthermore, a node is the purest if it has the records of merely one class.

$$Entropy = - \sum_i^n \log_2(P_i), \quad (1)$$

where n is the number of features, i is the feature, and P is the probability of i .

Entropy is computed for every feature, and the one providing the minimum value is chosen for the split. Thus, the mathematical range of entropy is from 0 to 1.

The next stage is to discover the information gain (IG); its values are between 0 and 1. A more considerable information gain suggests a lower entropy group of sample and hence less surprise. Moreover, information gain assists the tree decides that the feature split on, which provides the maximum information gain. Based on Equation (2), we can compute the information gained for every feature individually:

$$IG = Entropy(parent) - weighted_{avg} * Entropy(children). \quad (2)$$

Here, the parent is the target feature, and children are other features of the dataset. After calculating the information gain for each feature, sort all information gain in descending order. Now, we can select more prominent information gain features as optimal features.

5.3. Fisher Score-Based Feature Selection. Fisher score is Newton's technique utilized in statistics to solve maximum likelihood equations numerically [31]. For example, the score of the i th feature S_i will be computed through the Fisher score, which is shown as follows:

$$S_i = \frac{\sum n_j (u_{ij} - \mu_i)^2}{\sum n_j * \rho_{ij}^2}, \quad (3)$$

where μ_{ij} and ρ_{ij} are the mean and the variance of the i th feature in the j th class, correspondingly; n_j is the number of records in the j th class, and μ_i is the mean of the i th feature; after calculating the Fisher score for each feature, sort all Fisher scores based on descending order. Now, we can select more significant Fisher score features as optimal features.

5.4. Min-Max Normalization. One of the most common methods to normalize data is the min-max normalization [32]. First, for each feature, the minimum value of that feature gets modified into 0. Then, the maximum value gets modified into 1. In addition, each other value gets transmitted into a decimal between 0 and 1. Equation (4) shows the min-max normalization:

$$X_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}. \quad (4)$$

For instance, if the minimum value of a feature was $\min(x) = 20$, and the maximum value was $\max(x) = 40$, then $x = 30$ would be altered to $X_{norm} = 0.5$. Then, calculate the score for the i th feature using the average value of each instance in the i th feature; after calculating the min-max normalization score for each feature, sort all min-max normalization scores based on descending order. Now, we can select larger min-max normalization score features as the optimal features.

5.5. Correlation Coefficient. The correlation coefficient is a statistic utilized to calculate the linear correlation between feature X and the target Y feature [33]. It lies between +1 and -1, where 1 means a total positive correlation and -1 means an absolute negative correlation. Thus, 0 means that there is no linear correlation. To compute the correlation coefficient, get the input feature X and output feature Y 's covariance by dividing it by the product of the two features' standard deviation—the formula is

$$CC = \frac{Cov(X, Y)}{\sigma_X * \sigma_Y}, \quad (5)$$

where Cov is the covariance, and σ_X is the standard deviation of X ; also, σ_Y is the standard deviation of Y . Based on the above equation, the CC score of each feature can compute. After calculating the correlation coefficient score for each feature, sort all correlation coefficient scores based on descending order. Now, we can select more prominent correlation coefficient score features as the optimal features.

5.6. Mean Absolute Deviation. The mean absolute deviation of the dataset is the average distance between each data point and the mean [34]. Here is how to compute the average absolute deviation.

Step 1: Compute the mean

Step 2: Compute how far each data point is from the mean utilizing positive distance. These are known as absolute deviations

Step 3: Put those deviations jointly

Step 4: Separate the total through the number of data points

Following these stages in the instance below is perhaps the finest method to learn regarding the mean absolute deviation; however, here is a more formal method to write the steps into a formula which is shown as follows:

$$MAD = \frac{\sum |x_i - \bar{x}|}{n}. \quad (6)$$

Based on the above equation, the MAD score of each feature can compute after calculating the MAD score for each feature and sort all MAD scores based on descending order. We can select more significant MAD score features as optimal features. A subset of the dataset can be extracted based on these selected best features. It could also decrease the dimensionality of the dataset.

5.7. Novel Clustering Aggregation (NCA) Algorithm for Clustering Fall Detection Data. The accessibility of a massive set of clustering algorithms may confuse specialists attempting to choose the appropriate algorithm for the dataset [35]. Moreover, no clustering algorithm could universally solve all problems, for example, cluster shape, noise, or density [36]. To deal with this problem, this work proposed a Novel Clustering Aggregation (NCA) using the combination of three clustering algorithms, namely, K-means, expectation-maximization (EM), and density-based spatial clustering of applications with noise (DBSCAN). In the primary step, these three clustering algorithms cluster the general dataset separately. Then, the final clusters are taken out by a voting procedure between the data instances in the subsequent step. Finally, these three clustering algorithms assign data instances to the majority voted clusters. Thus, it enhances the accuracy of clustering and further decreases the clustering time compared to the unique clustering algorithms of the ensemble. Figure 6 shows the flow diagram of the NCA algorithm.

The following Algorithm 2 shows Novel Clustering Aggregation (NCA) algorithm.

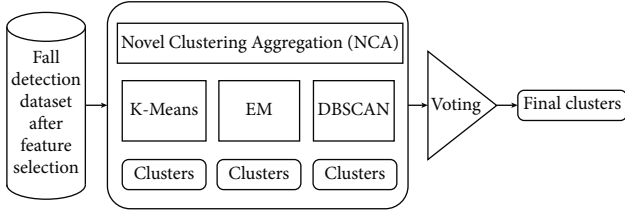


FIGURE 6: Flow diagram of the NCA algorithm.

5.8. K-Means Clustering. K-means clustering is the most standard partitioning clustering algorithm, a type of clustering that partitions the data into nonhierarchical groups [37]. It is alias centroid based clustering. In this algorithm, the dataset is separated into K sets, where K describes the number of predefined groups. The cluster's center is created so that the distance between the data points of one group is small compared to another cluster centroid [38].

K-means clustering objective is to divide n data instances into k clusters in which each data instance belongs to the group with the adjacent mean [39]. This algorithm creates precisely k different clusters. The goal of K-means clustering is to reduce the entire intracluster difference, or the squared error function is shown as follows:

$$P = \sum_{j=1}^k \sum_{i=1}^n |x_{ij} - c_j|^2. \quad (7)$$

where P is the objective function, n is the instance count, k is the clusters count, and $|x_{ij} - c_j|^2$ is the Euclidean distance.

5.9. Expectation-Maximization (EM) Clustering. The expectation-maximization algorithm, or EM algorithm for short, is the most familiar distribution model-based clustering algorithm, an approach for maximum likelihood estimation in latent variables [40]. The EM algorithm is an iterative approach to cycles between two modes [41]. The first mode attempts to estimate the missing or latent variables called the estimation step or E-step. The second mode attempts to optimize the model's parameters to explain the data, best called the maximization step or M-step.

- (i) E-Step: Estimate the missing variables in the dataset
- (ii) M-Step: Maximize the model's parameters in the presence of the data

The EM algorithm can be applied quite widely, although it is perhaps most well known in machine learning for use in unsupervised learning problems, such as density estimation and clustering.

5.10. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) Clustering. The DBSCAN clustering links highly-dense regions into clusters; arbitrarily shaped distributions were created as long as the crowded area could be connected [42]. This algorithm discovers various groups in the dataset and connects enormous densities' regions. Thus, the overcrowded areas of data space are separated from each

other through sparser areas. In a density-based clustering algorithm, points are categorized as reachable points, core points, and outliers, as follows:

- (i) A point y is accessible from x if there is a route x_1, \dots, x_n with $x_1 = x$ and $x_n = y$, where each x_{i+1} is straightly reachable from x_i
- (ii) A point x is a core point if the minimum points are within the distance ϵ (ϵ is the maximum radius of the neighborhood of x). Those points are straightly reachable from x . By definition, no points are straightly reachable from a noncore point
- (iii) All points that cannot reach from any other point are outliers

5.11. Majority Voting Technique. A majority vote is more than half of the votes cast. The plan behind the majority vote is that the verdict of a committee is higher than the verdict of individuals. The voting-based clustering technique is that each data instance in a specified dataset vote for the cluster it belongs to and its equivalent collection in each other clustering outcome. The highest of these values denotes the finest group for the data instance. It means that each data instance should cluster along with the opinion of the majority of the algorithms.

5.12. Novel Stacking Classification and Prediction Algorithm for Fall Detection Data. Stacking classification is an ensemble technique that merges multiple classifiers through a meta-classifier. The ensemble technique utilizes many classification algorithms to attain superior predictive performance than a single classification algorithm. Therefore, this paper proposed a novel stacking classification and prediction algorithm (NSCP) for fall detection data. In NSCP, we employed a stacking algorithm to classify the meta-features to attain the final class. Classifiers from the first layer (RIPPER + MLR + D14jMlpClassifier) return the probability of belonging to a class (meta-feature). In the second layer, these meta-features are the input of the meta-classifier (Naive Bayes classifier). Finally, the output of the classifier can be 0 (standing) or 1 (walking) or 2 (sitting) or 3 (falling) or 4 (cramps) or 5 (running). Figure 7 shows the flow diagram of the NSCP algorithm.

Algorithm 3 shows a novel stacking classification and prediction algorithm (NSCP).

5.13. Ripper Classifier. It refers to the Repeated Incremental Pruning to Produce Error Reduction. The RIPPER algorithm is one of the classification algorithms based on the rule. Furthermore, it derives a set of rules from the training dataset. Therefore, it is the most broadly utilized rule induction algorithm [43].

5.13.1. Ripper Algorithm Usage

- (1) It works well on data with unbalanced class partitions. In data, if we have multiple records, most of them are said to belong to a particular class, and if the rest of the records belong to different classes,

Input: Fall Detection Dataset (FDD)
Output: Assign each data instance to Majority Voted Cluster (MVC)
Step 1: Load Fall Detection Dataset with selected features
Step 2: Apply K-Means Clustering for FDD
Step 3: Apply EM Clustering for FDD
Step 4: Apply DBSCAN Clustering for FDD
Step 5: For each data instance DI from FDD
Step 6: Result1 = Get the K-Means cluster result for DI
Step 7: Result2 = Get the EM cluster result for DI
Step 8: Result3 = Get the DBSCAN cluster result for DI
Step 9: If(Result1 is equal to Result2), Then
Step 10: MVC = Result2
Step 11: Else If(Result1 is equal to Result3), Then
Step 12: MVC = Result3
Step 13: End if
Step 14: End For

ALGORITHM 2: Novel Clustering Aggregation (NCA)

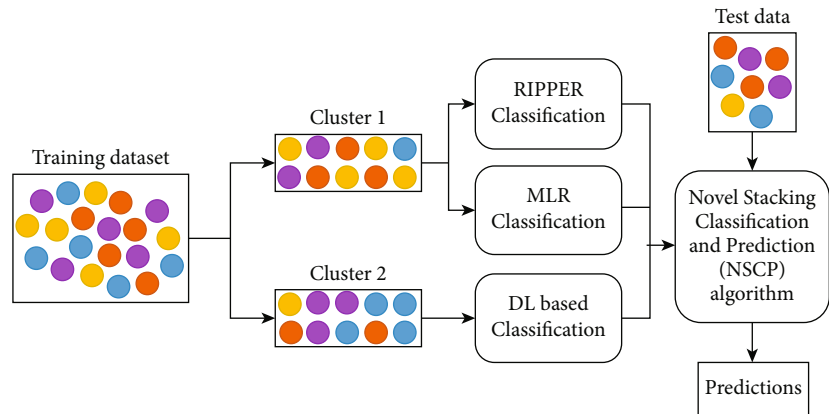


FIGURE 7: Flow diagram of the NSCP algorithm.

Input: Fall Detection Training Dataset Cluster 1 (FTC1), Fall Detection Training Dataset Cluster 2 (FTC2), Fall Detection Testing Dataset (FDTD)
Output: Fall Detection Predicted Result (FDPR)
Step 1: Classify FTC1 based on RIPPER classifier using weka
Step 2: Classify FTC1 based on MLR classifier using weka
Step 3: Classify FTC2 based on D14jMlpClassifier classifier using weka
Step 4: For each data instance DI from FDTD
Step 5: P1 = Predict DI using RIPPER classifier
Step 6: P2 = Predict DI using MLR classifier
Step 7: P3 = Predict DI using D14jMlpClassifier classifier
Step 8: FDPR = Predict DI using Stacking classifier (RIPPER + MLR + D14jMlpClassifier as base classifier and Naïve Bayes as Meta-Classifier)
Step 9: End For

ALGORITHM 3: Novel Stacking Classification and Prediction algorithm (NSCP)

TABLE 3: Comparison of accuracy among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS for fall detection dataset.

Algorithm	Accuracy (in %)
IG-based	79.48
FS-base	78.77
MMN-based	83.35
CC-based	79.09
MAD-based	87.72
MCFS	88.72

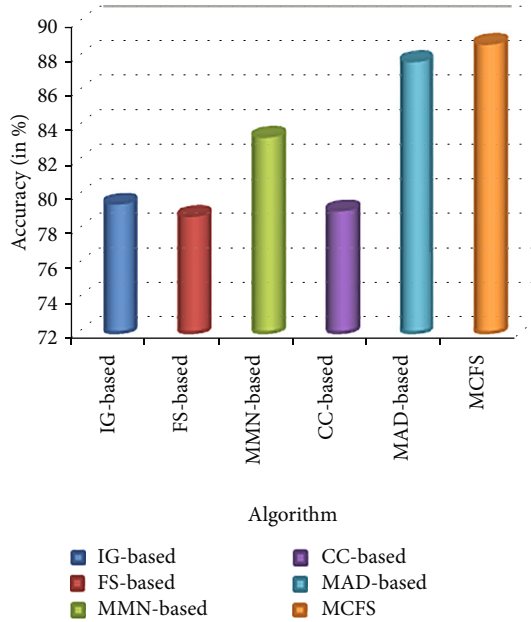


FIGURE 8: Accuracy comparison among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS.

the data is said to have an unbalanced class distribution

- (2) It performs better on noisy data by utilizing a verification set to avert the overfitting model

5.13.2. Functioning of Ripper Algorithm [44]

Case 1. Training records should be owned by merely two classes.

In the given records, it discovers the majority class (which seems to be the most) and gets this class as the default class. For instance, if there are 50 entries and 35 belong to Class A and 15 to Class B, Class A would subsequently be the default class. As for the other class, it attempts to learn/get different rules to find that class.

Case 2. There are more than two classes in the training records (numerous classes).

Regard the available classes and arrange them in a specific order based on their frequency.

Regard the classes are prearranged as follows:

TABLE 4: Comparison of sensitivity among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS for fall detection dataset.

Algorithm	Sensitivity
IG-based	0.8797
FS-base	0.76
MMN-based	0.8587
CC-based	0.8265
MAD-based	0.8693
MCFS	0.8897

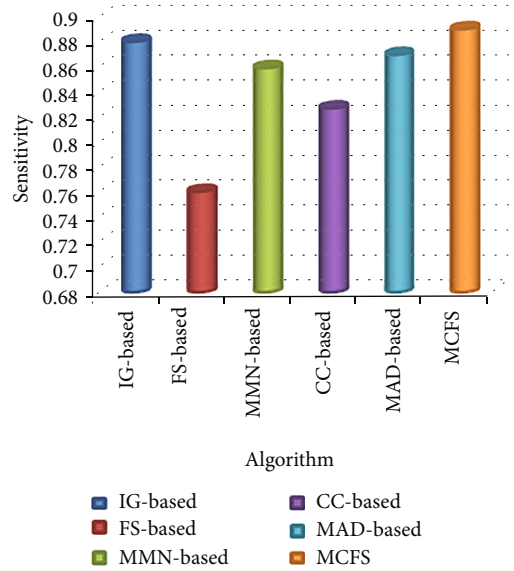


FIGURE 9: Comparison of sensitivity among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS.

- (i) $C_1, C_2, C_3, \dots, C_n$
- (ii) C_1 – minimum frequency
- (iii) C_n – maximum frequency

The most frequent (C_n) class is taken as the default class.

(1) *How the Rule Evolved.* In the primary case, it attempts to get the rules for C_1 class records. Entries belonging to C_1 will be treated as positive examples (+ve) and other classes as negative examples (-ve). Then, a sequential covering algorithm is utilized to create rules that discriminate between the +ve and -ve instances. After that, the RIPPER algorithm attempts to differentiate the rules for C_2 from other classes at this meeting. With C_n (default class) remaining, this procedure is continued until the stop criteria are met. Finally, the RIPPER algorithm separates the rules from the minority class to the majority class.

5.13.3. The Growing Rule in the RIPPER Algorithm

- (i) The RIPPER algorithm utilizes the standard to a particular plan of rising rules. It initiates with an

TABLE 5: Comparison of specificity among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS for fall detection dataset.

Algorithm	Specificity
IG-based	0.7821
FS-base	0.715
MMN-based	0.7908
CC-based	0.7248
MAD-based	0.7416
MCFS	0.8008

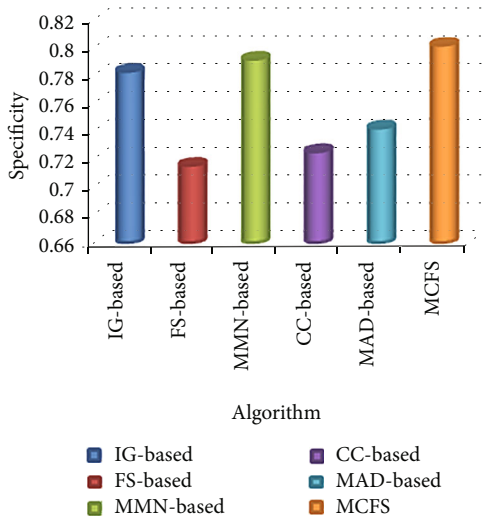


FIGURE 10: Comparison of specificity among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS.

TABLE 6: Comparison of precision among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS for fall detection dataset.

Algorithm	Precision
IG-based	0.7035
FS-base	0.7779
MMN-based	0.7471
CC-based	0.7878
MAD-based	0.8227
MCFS	0.8327

empty rule and continues to add the finest conjunct to the precursor of the rule

- (ii) Metric was selected for the evaluation of links is FOIL's information gain. Utilizing this, ideal link is selected
- (iii) Stopping criteria for adding conjuncts - when the rule starts covering the negative (-ve) examples
- (iv) The new rule is truncated using its performance in the set of validation

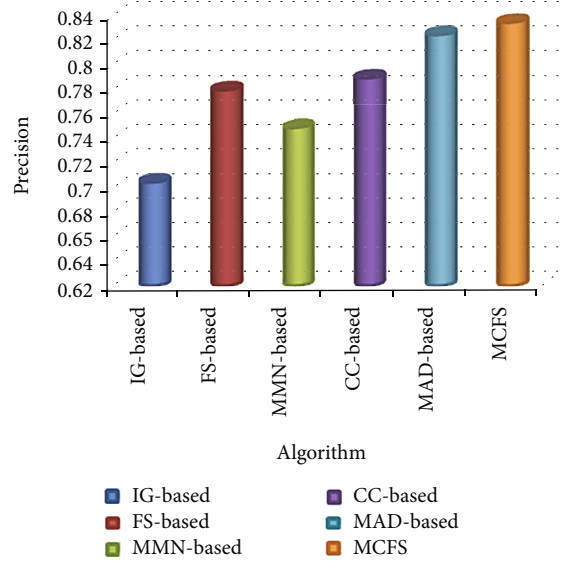


FIGURE 11: Comparison of precision among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS.

5.13.4. *Rule Pruning Using RIPPER Algorithm.* We require discovering whether or not a specific rule must be pruned. To decide, this metric is utilized, which is

$(P - N)/(P + N)$ where P is the number of +ve examples in the set of validations covered by the rule and N is the number of -ve examples in the set of validations covered by the rule.

- (i) Every time a conjunct is removed or attached, we compute the value of the above measurement for the original rule (before removing/adding) and the new rule (after removing/adding)
- (ii) If the value of the new rule is superior to the creative rule, we can remove/add the conjunct. Otherwise, the conjunct will not be eliminated/added
- (iii) Pruning is completed initiating from the rightmost end. For instance, regard a rule $-PQRS \rightarrow Z$, where $P, Q, R,$ and S are conjuncts and Z is the class

Initially, it will eliminate the conjunct S and compute the metric value. If the quality of the metric is enhanced, the conjunct S is eliminated. If the quality does not enhance, then the pruning is verified for $RS, QRS,$ etc.

5.13.5. *Creating a Rule Set in the RIPPER Algorithm*

- (i) Once a rule is obtained, all a +ve and -ve instances covered by the rule are removed
- (ii) The rule will then be added to the ruleset until the termination condition is violated. The stopping criteria which we could utilize are as follows:
 - (A) Minimum Descriptive Length Policy: To transfer data from one node to another, you need the least number of bits. We desire the rule to be specified utilizing the least number of bits.

TABLE 7: Comparison of execution time among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS for fall detection dataset.

Algorithm	Execution time (ms)
IG-based	6487
FS-base	30877
MMN-based	22815
CC-based	2358
MAD-based	15920
MCFS	3606

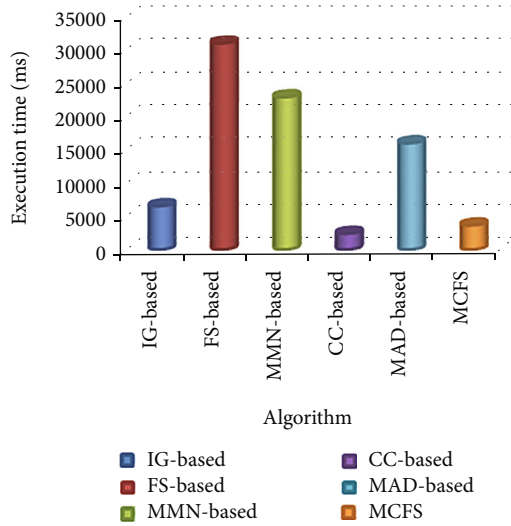


FIGURE 12: Comparison of execution time among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS.

If the new rule raises the total descriptive length of the ruleset through d bits (by default, d is 64 bits), then RIPPER stops adding rules to the set of rules

- (B) Error Ratio: We will review the rule and compute its error rate (incorrect classification) in the validation set. The error ratio of a specific rule must not exceed 50%.

5.14. MLR Classifier. Multinomial logistic regression (MLR) is a classification algorithm similar to logistic regression for binary classification [45]. In the logistic regression for binary classification, the classification work is to forecast the target class belonging to the binary type, for example, Yes or No, 0 or 1, and male or female. Regarding MLR, the plan utilizes logistic regression algorithms to forecast the target class (over two target classes).

As long as the probabilities for each target are calculated, the underlining method would be similar to the logistic regression for binary classification. Once the probabilities have been computed, convert them to a hot encoding, and calculate the exact optimal weight using cross-entropy techniques during the training procedure.

TABLE 8: Comparison of dataset size among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS for fall detection dataset.

Algorithm	Size of the dataset (KB)
Original dataset	60
IG-based	30
FS-base	35
MMN-based	30
CC-based	36
MAD-based	35
MCFS	29

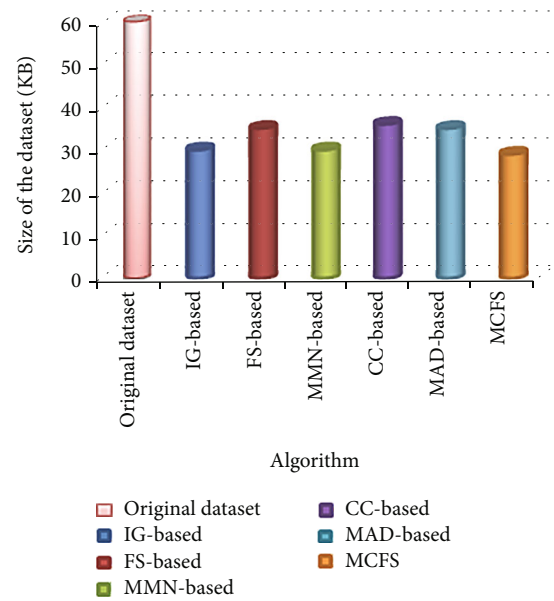


FIGURE 13: Comparison of size of the fall detection dataset among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS.

TABLE 9: Comparison of feature count among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS for fall detection dataset.

Algorithm	Number of features
Original dataset	7
IG-based	4
FS-base	4
MMN-based	3
CC-based	4
MAD-based	4
MCFS	3

5.14.1. Multinomial Logistic Regression Example. Using MLR, we can solve various kinds of classification issues. The trained model is utilized to forecast the target class from more than two target classes. Below are some instances to comprehend what type of issues we could solve utilizing MLR:

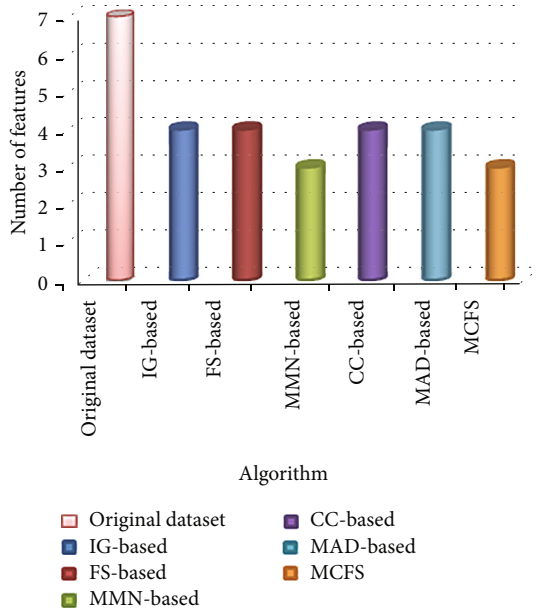


FIGURE 14: Comparison of features count among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS for fall detection dataset.

TABLE 10: Different clustering algorithms accuracy comparison for the fall detection dataset.

Algorithm	Accuracy (in %)
K-means clustering	50.67
EM clustering	97.46
DBSCAN clustering	77.64
NCA	97.88

(i) Forecasting the type of Iris flower species

Targets: a variety of species

(ii) Assessing the acceptability of the car utilizing the given attributes

Targets: very good, good, bad, and very bad

(iii) Forecasting the animal category utilizing the given animal attributes

Targets: camel, horse, cow, and deer

(1) *Advantages.*

(i) MLR is easy to execute and more proficient at interpreting and training

(ii) There is no assumption about the distribution of classes in the feature space

(iii) It not only presents a measure of how relevant a prediction (coefficient scale) is but also provides the direction (+ve or -ve) of its connection

(iv) It is more rapid in categorizing unknown records

(v) It works well for numerous easy datasets with good accuracy and when the dataset is linearly divisible

(vi) It could be interpreted as sample coefficients as an indicator of feature significance

6. D14jMlpClassifier

D14jMlpClassifier is one of the DL classification algorithms that arbitrarily create deep feedforward neural networks containing convolutional neural networks [46]. D14jMlpClassifier is the core technology of WekaDeeplearning4j, which is constructed from a Weka package, creating Deeplearning4j methods obtainable throughout the Weka tool. D14jMlpClassifier could be utilized for regression and classification by selecting suitable loss functions. The convolutional neural network (CNN) is a class of deep neural networks in deep learning. They are called space-inverting or shift-inverting artificial neural networks (SIANNs). They use the wide-weight configuration of convolution kernels or filters to slide on the input and present equivalent responses called hypothetical maps. Anti-intuitive, most CNNs are simply equivalent, unchanging, and contrary to translation. CNN regulates versions of multilayer perceptrons (MLPs). MLPs generally refer to entirely linked networks, that is, each neuron in one layer is linked to real neurons in the next layer. The “full connection” of these networks creates opportunities for data overload. Standard methods to avoid formalization or overfitting contain the following: penalize parameters by training or trimming attachments. CNN adopts various regularization techniques: taking advantage of the hierarchical pattern in information and increasing the difficulty of using minimal and easy patterns embedded in their filters. Therefore, even at the level of the connection problem, CNNs are inferior.

6.1. Stacking Classifier. Stacking classifier is a group technique in which the output of several classifiers is sent as input to a meta-classifier for final classification [47]. The stacking classifier technique is the most efficient way to implement multiple classification problems. Complete individual classification techniques, commonly known as basic learning techniques, can be integrated by creating a meta-classifier for the final result prediction task. It can be accomplished by stacking the results collectively from each classification algorithm and sending them like input to the meta-classifier. In the NSCP algorithm, the Naive Bayes classifier is used as a meta-classifier.

The Naïve Bayes algorithm is a supervised learning algorithm using the Bayes theorem that is also used to solve classification problems [48]. The Naïve Bayes classifier is one of the most accessible and efficient classification algorithms to help build rapid ML techniques to construct rapid predictions. It is a probability classifier, i.e., it forecasts the basis of an object’s probability. The Naïve Bayes algorithm consists of two terms, Naïve and Bayes, that could be explained as follows: It is known as Naïve since the frequency of a particular attribute is excluded from the frequency of other features. For example, if the fruit is recognized based on shape,

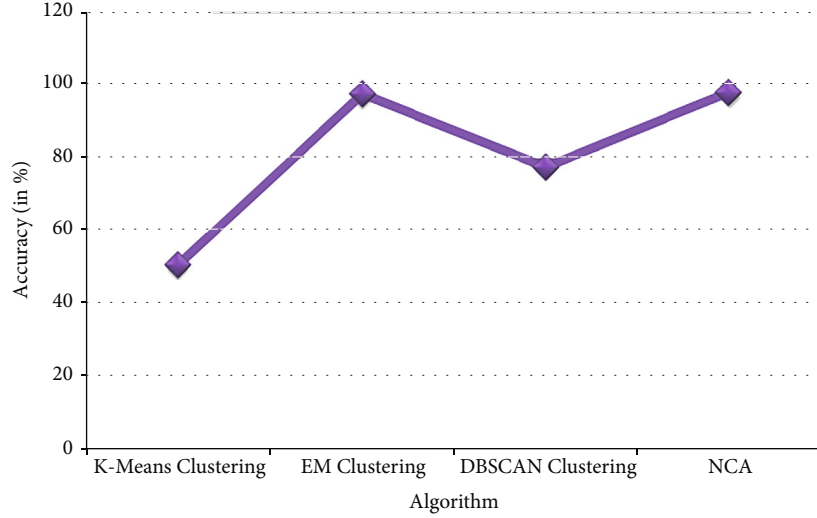


FIGURE 15: Different clustering algorithms accuracy comparison for fall detection dataset.

TABLE 11: Different clustering algorithms execution time comparison for the fall detection dataset.

Algorithm	Execution time (ms)
K-means clustering	2904
EM clustering	934
DBSCAN clustering	310
NCA	196

color, and taste, the red, spherical, and sweet fruits are identified as apples. Therefore, each attribute helps to recognize that it is an apple without trusting each other.

The analysis of dimensionality reduction techniques on big data was recommended in [49]. However, few authors worked on the hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis [50]. In [51], a meta-heuristic optimization approach for energy efficiency in the IoT networks. Hand gesture classification using a novel CNN-crow search algorithm was given by the authors in [52]. In healthcare data, an effective apriori approach for frequent pattern mining utilising mapreduce [53]. Covid-19 prediction using a recurrent neural network and reinforcement learning model [54]. In fog computing, an analysis of homomorphic techniques for data security [55]. In Bayes, it is called Bay because it relies on the principle of the Bayes theorem.

Bayes' theorem is called Bayes' law, which is utilized to decide the probability of a theory by preceding knowledge. This is because it relies on the probability of the condition.

7. Experimental Results

NSCP algorithms using MCFS with NCA algorithm are used to predict fall detection. Java and Weka tools both are used for algorithm implementation. The performances of these algorithms are assessed through accuracy, sensitivity, specificity, precision, recall, F-measure, execution time, dataset size, and the number of features.

7.1. Different Feature Selection Algorithms Comparison. To assess the effectiveness of the MCFS algorithm, we compared MCFS with other existing feature selection algorithms such as Information gain, Fisher score, min-max normalization, correlation coefficient, and mean absolute deviation based feature selection in terms of accuracy, sensitivity, specificity, precision, execution time, size of the dataset, and number of features.

7.1.1. Accuracy. Table 3 compares accuracy among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS dimensionality reduction algorithms.

Furthermore, Figure 8 shows an accuracy comparison. This comparison concludes MCFS algorithm is the best among others.

Among others, the accuracy of the FS-based algorithm is significantly less. However, the CC-based algorithm presents the highest accuracy compared with the FS-based algorithm. However, the IG-based algorithm presents the highest accuracy compared with the CC-based algorithm. However, compared with the IG-based algorithm, the accuracy of the MMN-based algorithm is very high. However, compared with the MMN-based algorithm, the MAD-based algorithm presents the highest accuracy. However, compared with the MAD-based algorithm, the accuracy of the MCFS algorithm is very high.

7.1.2. Sensitivity. Table 4 compares sensitivity among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS dimensionality reduction algorithms.

Furthermore, Figure 9 shows a sensitivity comparison. This comparison concludes MCFS algorithm is the best among others.

Among others, the sensitivity of the FS-based algorithm is significantly less. However, the CC-based algorithm presents the highest sensitivity compared with the FS-based algorithm. However, the MMN-based algorithm presents the most heightened sensitivity compared with the CC-based algorithm. However, compared with the MMN-

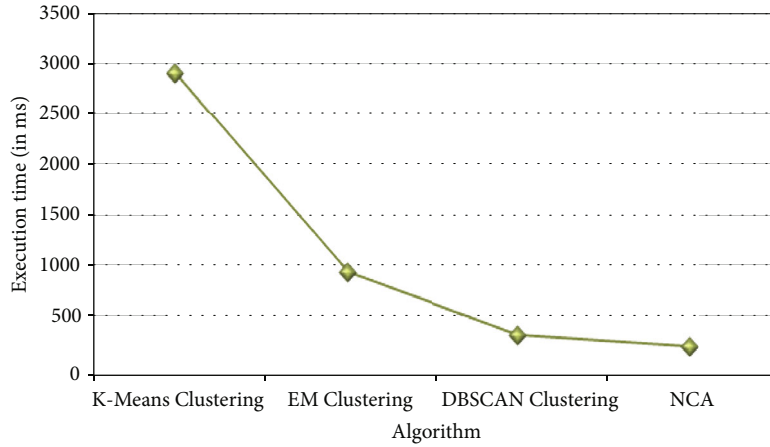


FIGURE 16: Execution time comparison of different clustering algorithms for fall detection dataset.

TABLE 12: Accuracy comparison of different classification algorithms.

Algorithm	Accuracy (in %)
RIPPER	84
MLR	87
Dl4jMlpClassifier	92
NSCP	98

TABLE 13: Precision comparison of RIPPER, MLR, Dl4jMlpClassifier, and NSCP algorithms.

Algorithm	Precision
RIPPER	0.85
MLR	0.88
Dl4jMlpClassifier	0.94
NSCP	0.96

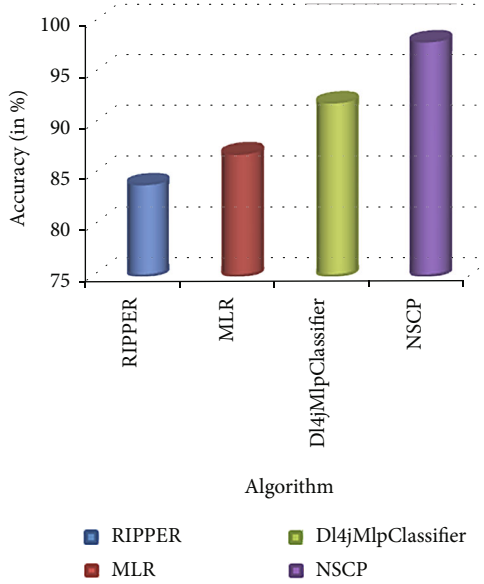


FIGURE 17: Accuracy comparisons of RIPPER, MLR, Dl4jMlpClassifier, and NSCP algorithms.

based algorithm, the sensitivity of the MAD-based algorithm is very high. However, the IG-based algorithm presents the most heightened sensitivity compared with the MAD-based algorithm. However, compared with the IG-based algorithm, the sensitivity of the MCFS algorithm is very high.

7.1.3. Specificity. Table 5 compares specificity among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS dimensionality reduction algorithms.

Furthermore, Figure 10 shows a specificity comparison. This comparison concludes MCFS algorithm is the best among others.

Among others, the specificity of the FS-based algorithm is significantly less. However, the CC-based algorithm presents the highest specificity compared with the FS-based algorithm. However, the MAD-based algorithm presents the highest specificity compared with the CC-based algorithm. However, compared with the MAD-based algorithm, the specificity of the IG-based algorithm is very high. However, the MMN-based algorithm presents the highest specificity compared with the IG-based algorithm. However, compared with the MMN-based algorithm, the specificity of the MCFS algorithm is very high.

7.1.4. Precision. Table 6 compares precision among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS dimensionality reduction algorithms.

Furthermore, Figure 11 shows the precision comparison. This comparison concludes MCFS algorithm is the best among others.

Among others, the precision of the IG-based algorithm is significantly less. However, the MMN-based algorithm presents the highest precision compared with the IG-based algorithm. However, compared with the MMN-based algorithm, the FS-based algorithm presents the highest precision. However, compared with the FS-based algorithm, the precision of the CC-based algorithm is very high. However, the MAD-based algorithm presents the highest precision compared with the CC-based algorithm. However, compared

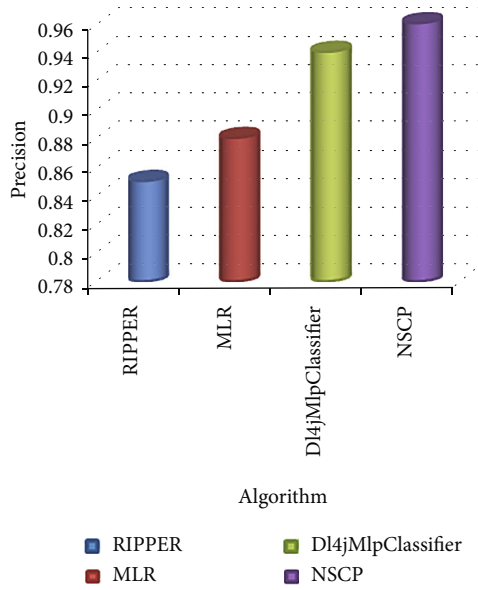


FIGURE 18: Precision comparisons of RIPPER, MLR, D14jMlpClassifier, and NSCP algorithms.

TABLE 14: Recall comparison of RIPPER, MLR, D14jMlpClassifier, and NSCP algorithms.

Algorithm	Recall
RIPPER	0.83
MLR	0.82
D14jMlpClassifier	0.93
NSCP	0.95

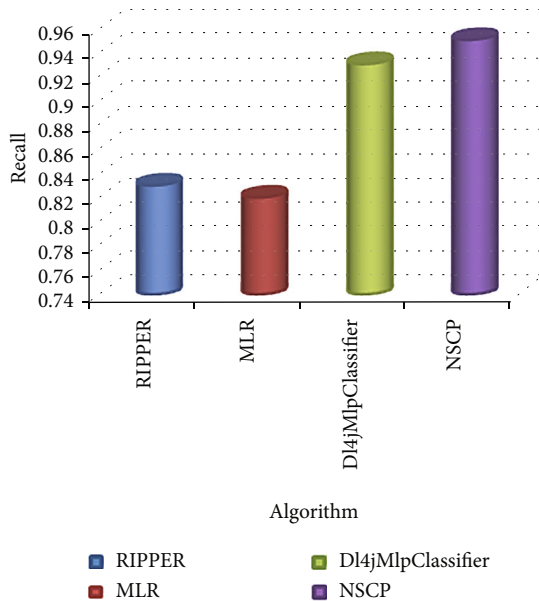


FIGURE 19: Recall comparisons of RIPPER, MLR, D14jMlpClassifier, and NSCP algorithms.

TABLE 15: F-Measure comparison of RIPPER, MLR, D14jMlpClassifier, and NSCP algorithms.

Algorithm	F-measure
RIPPER	0.80
MLR	0.88
D14jMlpClassifier	0.92
NSCP	0.98

with the MAD-based algorithm, the precision of the MCFS algorithm is very high.

7.1.5. Execution Time. Table 7 compares execution time among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS dimensionality reduction algorithms.

Furthermore, Figure 12 shows an execution time comparison. This comparison concludes that the CC-based algorithm is the best among others.

Among others, the execution time of the FS-based algorithm is very high. However, compared with the FS-based algorithm, the MMN-based algorithm provides a lower execution time. However, compared with the MMN-based algorithm, the MAD-based algorithm provides a lower execution time. However, compared with the MAD-based algorithm, the execution time of the IG-based algorithm is significantly less. However, the MCFS algorithm provides a lower execution time compared with the IG-based algorithm. However, compared with the MCFS algorithm, the execution time of the CC-based algorithm is significantly less.

7.1.6. Size of the Dataset. Table 8 compares the size of the dataset among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS dimensionality reduction algorithms.

Furthermore, Figure 13 shows the size of the dataset comparison. This comparison concludes that the MAD algorithm is the best among others.

The fall detection dataset size is 60 KB. After dimensionality reduction, the size of the dataset is compared here. Among others, the size of the dataset of the CC-based algorithm is enormous. Furthermore, the FS-based and MAD-based algorithms present a smaller dataset size than the CC-based algorithm. However, the IG-based and MMN-based algorithm presents a smaller dataset size compared with the FS-based and MAD-based algorithms. However, compared with the IG-based and MMN-based algorithms, the size of the dataset of the MCFS algorithm is significantly less.

7.1.7. The Number of Features. Table 9 compares feature count among IG-based, FS-based, MMN-based, CC-based, MAD-based, and MCFS dimensionality reduction algorithms.

Furthermore, Figure 14 shows several feature comparisons. This comparison concludes MCFS algorithm is the best among others.

The fall detection dataset has seven features. After dimensionality reduction, several features are compared here. Among others, the number of features of the IG-based,

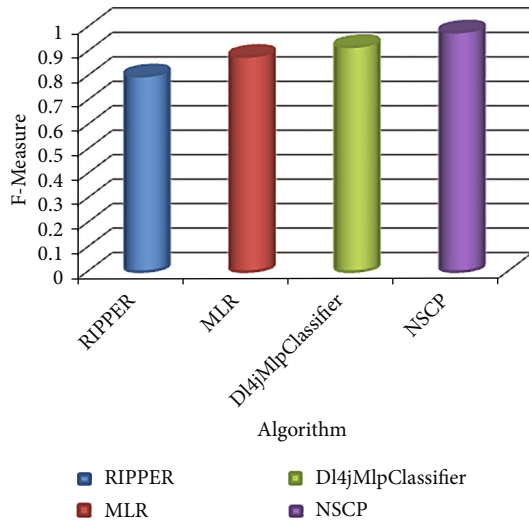


FIGURE 20: F-measure comparisons of RIPPER, MLR, D14jMlpClassifier, and NSCP algorithms.

FS-based, CC-based, and MAD-based algorithms is very high. Furthermore, the MMN-based algorithm and the MCFS algorithm provide fewer features compared with the others.

7.2. Different Clustering Algorithms Comparison. To evaluate the performance of the NCA algorithm, compare NCA with other existing clustering algorithms such as K-means clustering, EM clustering, and DBSCAN clustering in terms of accuracy and execution time.

7.2.1. Accuracy. Table 10 shows an accuracy comparison of different clustering algorithms using the fall detection dataset.

Furthermore, Figure 15 demonstrates the accuracy comparison for the fall detection dataset. Compared with other types of clustering proposed, NCA accuracy is high.

7.2.2. Execution Time. Table 11 shows the execution time comparison of the different clustering algorithms for the fall detection dataset.

Furthermore, Figure 16 demonstrates the comparison of execution time for the fall detection dataset. Compared with other types of clustering proposed, NCA takes less time for clustering.

7.3. Different Classification Algorithms Comparison. To evaluate the performance of the NSCP algorithm, compare NSCP with other existing classification algorithms such as RIPPER, MLR, and D14jMlpClassifier in terms of accuracy, precision, recall, and f-measure.

7.3.1. Accuracy. Table 12 shows the accuracy comparison of RIPPER, MLR, D14jMlpClassifier, and NSCP algorithms.

Figure 17 demonstrates the comparison of accuracy for the fall detection dataset. Compared with other classification algorithms, the accuracy of the NSCP algorithm is highest. The NSCP algorithm is used using the ensemble and stacking classifier approach. This approach boosts the perfor-

mance of the NSCP algorithm. Therefore, the NSCP algorithm provides the highest accuracy.

7.3.2. Precision. Table 13 demonstrates the precision comparison of the RIPPER, MLR, D14jMlpClassifier, and NSCP algorithms.

Figure 18 demonstrates the comparison of precision for the fall detection dataset. Compared with other classification algorithms, the precision of the proposed NSCP algorithm is enormous. Both ML (RIPPER, MLR, Nave Bayes) and DL (D14jMlpClassifier) methods are employed because the NSCP algorithm is used. Here, DL provides high accuracy. However, ML gives lesser accuracy than DL. Additionally, DL requires enormous data. However, ML can train with more minor data. By this, we can know that ML solves the defect of DL and DL solves the defect of ML. The NSCP algorithm uses both techniques, which provides the highest precision.

7.3.3. Recall. Table 14 shows the recall comparison of RIPPER, MLR, D14jMlpClassifier, and NSCP algorithms.

Figure 19 demonstrates the comparison of recall for the fall detection dataset. Compared with RIPPER, MLR, and D14jMlpClassifier algorithms, recall of the NSCP algorithm is high.

7.3.4. F-Measure. Table 15 demonstrates the F-measure comparison of RIPPER, MLR, D14jMlpClassifier, and NSCP algorithms.

Figure 20 demonstrates the comparison of F-Measure for the fall detection dataset. Compared with other classification algorithms, F-Measure of the proposed NSCP algorithm is high.

8. Conclusions

Falling is a rather common occurrence among older individuals, and it can have serious health consequences. Falls can result in physical ailments such as fractures, head traumas, and tooth decay. Falls will significantly impact specific populations, motivating the quest for better ways to prevent and respond to falls. As a result, using Multi-strategy Combination based Feature Selection (MCFS) and Novel Clustering Aggregation (NCA) algorithms, this research developed a Novel Stacking Classification and Prediction (NSCP) algorithm based on AAL for the elderly. The major goal of this study is to recognize older people's activities such as standing, walking, sitting, falling, cramps, and jogging. The experimental result shows that the proposed MCFS algorithm provides the highest accuracy, sensitivity, specificity, and precision; provides less execution time; reduces dataset size; and reduces the number of features. In addition, the NCA algorithm provided the highest accuracy and took less execution time than the three existing clustering algorithms. It concluded that the NSCP algorithm predicts fall detection efficiently. The NSCP algorithm, on the other hand, mixes ML and DL, which consumes more time and space. To deal with this difficulty in the future, improved NSCP algorithm (INSCP) will be required.

Data Availability

The data supporting this study's findings are unavailable in any public repositories.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Jinesh Padikkapparambil was assigned in the conceptualization, methodology, and writing. Cornelius Ncube was assigned in the methodology. Firoz Khan, Lakshmana Kumar Ramasamy, and Yomiyu Reta Gashu Dev were assigned in writing and editing.

Acknowledgments

This work will be funded by the Tepi University, Addis Ababa, Ethiopia.

References

- [1] D. Howdon and N. Rice, "Health care expenditures, age, proximity to death and morbidity: implications for an ageing population," *Journal of Health Economics*, vol. 57, pp. 60–74, 2018.
- [2] M. S. Kelgeri and R. M. Desai, "A study on health problems among the elderly," *Asian Journal of Home Science*, vol. 15, no. 1, pp. 44–48, 2020.
- [3] X. Zhao, D. Zhang, M. Wu et al., "Loneliness and depression symptoms among the elderly in nursing homes: a moderated mediation model of resilience and social support," *Psychiatry Research*, vol. 268, pp. 143–151, 2018.
- [4] H. Kim, Y. H. Park, Y. I. Jung et al., "Evaluation of a technology-enhanced integrated care model for frail older persons: protocol of the SPEC study, a stepped-wedge cluster-randomized trial in nursing homes," *BMC Geriatrics*, vol. 17, no. 1, pp. 1–14, 2017.
- [5] H. Ghayvat, S. Mukhopadhyay, B. Shenjie, A. Chouhan, and W. Chen, "Smart home-based ambient assisted living: recognition of anomaly in the activity of daily living for an elderly living alone. In 2018 IEEE international instrumentation and measurement technology conference (I2MTC) (pp. 1-5). IEEE," 2018.
- [6] L. Syed, S. Jabeen, S. Manimala, and A. Alsaedi, "Smart healthcare framework for ambient assisted living using IoMT and big data analytics techniques," *Future Generation Computer Systems*, vol. 101, pp. 136–151, 2019.
- [7] M. N. Alkhomsan, M. A. Hossain, S. M. M. Rahman, and M. Masud, "Situation awareness in ambient assisted living for smart healthcare," *IEEE Access*, vol. 5, pp. 20716–20725, 2017.
- [8] D. Loza-Matovelle, A. Verdugo, E. Zalama, and J. Gómez-García-Bermejo, "An architecture for the integration of robots and sensors for the care of the elderly in an ambient assisted living environment," *Robotics*, vol. 8, no. 3, p. 76, 2019.
- [9] G. Diraco, A. Leone, and P. Siciliano, "A radar-based smart sensor for unobtrusive elderly monitoring in ambient assisted living applications," *Biosensors*, vol. 7, no. 4, p. 55, 2017.
- [10] G. Marques, "Ambient assisted living and the internet of things. Harnessing the internet of everything (IoE) for accelerated innovation opportunities, 100-115," 2019.
- [11] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.
- [12] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," *Artificial Intelligence Review*, vol. 53, no. 2, pp. 907–948, 2020.
- [13] B. Remeseiro and V. Bolon-Canedo, "A review of feature selection methods in medical applications," *Computers in Biology and Medicine*, vol. 112, p. 103375, 2019.
- [14] A. Saxena, M. Prasad, A. Gupta et al., "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, 2017.
- [15] D. Kaur and J. Kaur, "Data Mining in Supermarket: A Survey," *International Journal of Computational Intelligence Research*, vol. 13, no. 8, pp. 1945–1951, 2017.
- [16] E. Aljalbout, V. Golkov, Y. Siddiqui, M. Strobel, and D. Cremers, "Clustering with deep learning: taxonomy and new methods," 2018, <http://arxiv.org/abs/1801.07648>.
- [17] R. Bala and D. Kumar, "Classification using ANN: A review," *International Journal of Computational Intelligence Research*, vol. 13, no. 7, pp. 1811–1820, 2017.
- [18] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modelling on imbalanced domains," *Computing Surveys*, vol. 49, no. 2, pp. 1–50, 2016.
- [19] M. S. Amin, Y. K. Chiam, and K. D. Varathan, "Identification of significant features and data mining techniques in predicting heart disease," *Telematics and Informatics*, vol. 36, pp. 82–93, 2019.
- [20] Y. Liu, Y. Mu, K. Chen, Y. Li, and J. Guo, "Daily activity feature selection in smart homes based on Pearson correlation coefficient," *Neural Processing Letters*, vol. 51, no. 2, pp. 1771–1787, 2020.
- [21] A. M. Helmi, M. A. Al-Qaness, A. Dahou, R. Damaševičius, T. Krilavičius, and M. A. Elaziz, "A novel hybrid gradient-based optimizer and grey wolf optimizer feature selection method for human activity recognition using smartphone sensors," *Entropy*, vol. 23, no. 8, p. 1065, 2021.
- [22] N. D. Nguyen, D. T. Bui, P. H. Truong, and G. M. Jeong, "Position-based feature selection for body sensors regarding daily living activity recognition," *Journal of Sensors*, vol. 2018, 13 pages, 2018.
- [23] A. Manzi, P. Dario, and F. Cavallo, "A human activity recognition system based on dynamic clustering of skeleton data," *Sensors*, vol. 17, no. 5, p. 1100, 2017.
- [24] F. Cruciani, C. D. Nugent, J. M. Quero et al., "Personalizing activity recognition with a clustering based semi-population approach," *IEEE Access*, vol. 8, pp. 207794–207804, 2020.
- [25] M. Fañez, J. R. Villar, E. de la Cal, V. M. González, and J. Sedano, "Feature clustering to improve fall detection: a preliminary study," in *International Workshop on Soft Computing Models in Industrial and Environmental Applications*, pp. 219–228, Springer, Cham, 2019.
- [26] H. Li, A. Shrestha, F. Fioranelli et al., "Multi-sensor data fusion for human activities classification and fall detection," in *2017 IEEE SENSORS (pp. 1-3)*, Glasgow, UK, October 2017.
- [27] A. Chelli and M. Pätzold, "A machine learning approach for fall detection and daily living activity recognition," *IEEE Access*, vol. 7, pp. 38670–38687, 2019.

- [28] D. Yacchirema, J. S. de Puga, C. Palau, and M. Esteve, "Fall detection system for older adults using IoT and ensemble machine learning algorithm," *Personal and Ubiquitous Computing*, vol. 23, no. 5-6, pp. 801-817, 2019.
- [29] "Fall Detection Data from China," <https://www.kaggle.com/pitasr/falldata>.
- [30] S. Jadhav, H. He, and K. Jenkins, "Information gain directed genetic algorithm wrapper feature selection for credit rating," *Applied Soft Computing*, vol. 69, pp. 541-553, 2018.
- [31] D. Aksu, S. Üstebay, M. A. Aydin, and T. Atmaca, "Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm," in *International Symposium on Computer and Information Sciences*, pp. 141-149, Springer, Cham, 2018.
- [32] L. Munkhdalai, T. Munkhdalai, K. H. Park, H. G. Lee, M. Li, and K. H. Ryu, "mixture of activation functions with extended min-max normalization for forex market prediction," *IEEE Access*, vol. 7, pp. 183680-183691, 2019.
- [33] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaei, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *Journal of Information Security and Applications*, vol. 44, pp. 80-88, 2019.
- [34] U. Khair, H. Fahmi, S. Al Hakim, and R. Rahim, "Forecasting error calculation with mean absolute deviation and mean absolute percentage error," *Journal of Physics: Conference Series*, vol. 930, no. 1, article 012002, 2017.
- [35] A. Rosato, R. Altילו, and M. Panella, "A decentralized algorithm for distributed ensemble clustering," *Information Sciences*, vol. 578, pp. 417-434, 2021.
- [36] K. Bindra, A. Mishra, and Suryakant, "Effective data clustering algorithms," in *Soft Computing: Theories and Applications*, pp. 419-432, Springer, Singapore, 2019.
- [37] K. P. Sinaga and M. S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716-80727, 2020.
- [38] C. Yuan and H. Yang, "Research on K-value selection method of K-means clustering algorithm," *Multidisciplinary Scientific Journal*, vol. 2, no. 2, pp. 226-235, 2019.
- [39] S. Chakraborty, D. Paul, S. Das, and J. Xu, "Entropy weighted power k-means clustering," in *International Conference on Artificial Intelligence and Statistics*, pp. 691-701, PMLR, 2020.
- [40] N. Sammaknejad, Y. Zhao, and B. Huang, "A review of the expectation-maximization algorithm in data-driven process identification," *Journal of Process Control*, vol. 73, pp. 123-136, 2019.
- [41] A. S. Şimşek and H. Topaloglu, "An expectation-maximization algorithm to estimate the parameters of the Markov chain choice model," *Operations Research*, vol. 66, no. 3, pp. 748-760, 2018.
- [42] S. F. Galán, "Comparative evaluation of region query strategies for DBSCAN clustering," *Information Sciences*, vol. 502, pp. 76-90, 2019.
- [43] H. He, X. Zhang, Q. Wang et al., "Ensemble multi boost based on ripper classifier to predict imbalanced software defect data," *IEEE Access*, vol. 7, pp. 110333-110343, 2019.
- [44] <https://www.geeksforgeeks.org/ripper-algorithm/>.
- [45] K. Kayabol, "Approximate sparse multinomial logistic regression for classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 490-493, 2020.
- [46] S. Lang, F. Bravo-Marquez, C. Beckham, M. Hall, and E. Frank, "Wekadeeplearning4j: a deep learning package for weka based on deeplearning4j," *Knowledge-Based Systems*, vol. 178, pp. 48-50, 2019.
- [47] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one-class support vector machine," *Electronics*, vol. 9, no. 1, p. 173, 2020.
- [48] N. Salmi and Z. Rustam, "Naïve Bayes classifier models for predicting colon cancer," *IOP Conference Series: Materials Science and Engineering*, vol. 546, no. 5, article 052068, 2019IOP Publishing.
- [49] G. T. Reddy, M. P. K. Reddy, K. Lakshmana et al., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776-54788, 2020.
- [50] G. T. Reddy, M. Reddy, K. Lakshmana, D. S. Rajput, R. Kaluri, and G. Srivastava, "Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis," *Evolutionary Intelligence*, vol. 13, no. 2, pp. 185-196, 2020.
- [51] C. Iwendi, P. K. Maddikunta, T. R. Gadekallu, K. Lakshmana, A. K. Bashir, and M. J. Piran, "A metaheuristic optimization approach for energy efficiency in the IoT networks," *Software: Practice and Experience*, vol. 51, no. 12, pp. 2558-2571, 2021.
- [52] T. R. Gadekallu, M. Alazab, R. Kaluri, P. K. Maddikunta, S. Bhattacharya, and K. Lakshmana, "Hand gesture classification using a novel CNN-crow search algorithm," *Complex & Intelligent Systems*, vol. 7, no. 4, pp. 1855-1868, 2021.
- [53] M. Sornalakshmi, S. Balamurali, M. Venkatesulu et al., "An efficient apriori algorithm for frequent pattern mining using mapreduce in healthcare data," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 390-403, 2021.
- [54] R. L. Kumar, F. Khan, S. Din, S. S. Band, A. Mosavi, and E. Ibeke, "Recurrent neural network and reinforcement learning model for covid-19 prediction," *Frontiers in Public Health*, vol. 9, 2021.
- [55] A. Murugesan, B. Saminathan, F. Al-Turjman, and R. L. Kumar, "Analysis on homomorphic technique for data security in fog computing," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 9, article e3990, 2021.