Computing and Informatics, Vol. 44, 2025, 1101–1122, doi: 10.31577/cai_2025_5_1101

DESIGN AND DEVELOPMENT OF A HYBRID EVOLUTIONARY METHOD WITH A SPECIAL SELECTION OF ARTIFICIAL IMMUNE SYSTEM FOR STROKE PREDICTION: A BALANCING APPROACH

Şerife Çelikbaş

Biomedical Device Technology
İstanbul Aydın University
34315 İstanbul, Türkiye
e-mail: serifecelikbas@aydin.edu.tr, serifecelikbas@gmail.com

Zeynep Orman

Computer Engineering
İstanbul University-Cerrahpaşa
34315 İstanbul, Türkiye
e-mail: ormanz@iuc.edu.tr

Abdurrahim Akgündoğdu

Electrical Electronics Engineering
İstanbul University-Cerrahpaşa
34315 İstanbul, Türkiye
e-mail: akgundogdu@iuc.edu.tr

Abstract. A stroke is a serious neurological condition that occurs due to either blockages or bleeding in the brain, which can lead to death or long-term disability. This study aims to enhance the accuracy of disease diagnosis in imbalanced stroke patient datasets. The model incorporates an artificial immune system algorithm, whose parameters are fine-tuned using the Firefly algorithm to ensure both stability and balanced data. To enhance the performance of the underrepresented class, the One-Sided Selection method is employed. The model's effectiveness was tested in two separate experiments: one utilizing all available features and the other applying the Artificial Bee Colony (ABC) algorithm to select the most relevant features. The models were trained using six different classification algorithms: CatBoost, Light Gradient Boosting Machine (LightGBM), Gradient Boosting (GB), Extreme Gradient Boosting (XGBoost), Support Vector Machine (SVM), and Logistic Regression (LR). The results were presented using performance metrics. When trained using all features, the model achieved an accuracy of 93 %, specificity of 93 %, and sensitivity of 80 %. When trained using the best features selected by the ABC algorithm, the model achieved an accuracy of 93 %, specificity of 93 %, and sensitivity of 82 %. Compared to previous studies, the proposed model was effective in both experiments.

Keywords: Stroke disease prediction, artificial immune system, imbalanced dataset, one-sided selection

1 INTRODUCTION

Stroke is a neurological disorder resulting from vascular obstruction or hemorrhage in the brain, and its prevalence has been rising in recent years [1, 2]. As the disease can lead to physical disabilities and long hospitalization periods, it can significantly affect patients [3]. Early diagnosis is crucial to take necessary precautions against the risks associated with the disease. However, traditional diagnostic methods, such as computerized tomography, can be costly and time-consuming, making them less efficient [4]. Therefore, machine learning techniques have been widely used in recent years as a faster and more cost-effective way of diagnosing diseases, including stroke [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]. A diverse range of studies has utilized artificial intelligence algorithms and machine learning techniques to diagnose stroke. For instance, Arslan et al. [5] applied three different data mining approaches—Support Vector Machine (SVM), Stochastic Gradient Boost (SGB), and Penalized Logistic Regression (PLR) – to a dataset consisting of 80 patients and 112 healthy individuals. The results were satisfactory for the SVM and SGB models. Similarly, Puspitasari et al. [6] used the "Stroke Prediction Dataset" from Kaggle, containing 5 110 records, to predict stroke risk. Their findings showed that the Random Forest (RF) algorithm outperformed the Decision Tree (DT) in accuracy.

One of the main challenges in using machine learning for disease detection is dataset imbalance. This happens when there are significantly more samples of one class (such as unhealthy individuals) than the other, often due to incomplete patient information or uneven colletion of collected healthy versus diseased samples [7, 8, 9]. As a result, machine learning algorithms tend to focus on achieving high accuracy for the majority class, reducing their effectiveness effectiveness in detecting the minority class – those with the disease [10]. Dataset imbalance is a widely discussed issue in machine learning-based disease detection studies, and also poses challenges for stroke diagnosis. Past studies have suggested solutions for missing data, unbalanced datasets, and feature selection in stroke disease detection [21, 22, 23].

To achieve efficient and accurate stroke diagnosis, several swarm intelligence algorithms have been employed with varying degrees of success in previous studies [15, 16]. For example, Yagin et al. [11] applied the gradient boosting tree classification method to an unbalanced dataset of 5 110 samples from the open-access Kaggle dataset. To handle the imbalance, they used the Synthetic Minority Over-Sampling Technique (SMOTE), which produced more realistic results and highlighted the importance of balancing in stroke diagnosis preprocessing. Similarly, Sailasya and Kumari [12] tested several machine learning algorithms – Logistic Regression (LR), DT, RF, K-Nearest Neighbors (KNN), SVM, and Naive Bayes – for stroke prediction using the same dataset. They addressed missing data through mean value imputation and dataset imbalance using undersampling, with Naive Bayes delivering the best performance. Other studies have also tackled similar challenges. Rana et al. [13] used KNN imputation for missing data and SMOTE-Tomek for balancing the dataset, finding that artificial intelligence yielded the best results among the classification algorithms tested. Dev and Malik [14] utilized the "Stroke Prediction Dataset," consisting of 43 400 records, and addressed missing data by averaging. Their work employed the Artificial Bee Colony (ABC) algorithm for feature selection, achieving more accurate predictions. Liu et al. [15] solved the missing data problem using the RF algorithm, applying Principal Component Analysis (PCA) and K-means clustering for balancing. Santos et al. [16] used the same dataset as Liu et al. and addressed data imbalance using the artificial immune algorithm, optimized according to the Kohonen network for oversampling and hyperparameter tuning. They employed decision trees generated by the genetic programming algorithm (DT-GP) in the classification step, achieving satisfactory performance.

This study focuses on the problem of data imbalance, which is frequently encountered in datasets used for the diagnosis of stroke disease, and the resulting adverse effects on the performance of classification methods. In order to provide a solution to this problem, it proposes an evolutionary data balancing approach-based balancing model. The model uses the Clonal Selection Algorithm (ClonALG) combined with the One-Sided Selection (OSS) model and enhanced with the Firefly Algorithm (FFA) to balance the data. As a result, it has been shown that effective results are obtained by comparing the performances with various classification methods with current studies in the literature. In summary, the contributions of this study can be given as follows:

- A hybrid evolutionary model-based data balancing approach is proposed, which differs from methods used in the literature, for predicting stroke using an incomplete and unstable medical dataset.

- In the Artificial Immune System (AIS), sufficient numbers and varieties of antibodies are generated using a selection algorithm updated according to FFA's attractiveness. This contributes positively to data balancing by increasing data diversity. The high mutation diversity further ensures effective performance results.
- The mutation data modeled according to the OSS algorithm increased the patient data rate from 1.8 % to at least 15.5 % for each training dataset created in the cross-fold, providing effective results in learning rates.
- For the data balancing problem in the detection of stroke disease, the same data set was compared with the studies in the literature and it was shown that effective comparable results were obtained in performance metrics.
- Unlike the literature studies using the same data set, feature selection was made according to the ABC algorithm which positively affected the performance metrics.

The rest of the paper is organized as follows: Section 2 introduces the dataset and methods used. Section 3 describes the operation of the proposed balancing model and its algorithm structure. Section 4 presents experimental results and comparative analyses, and Section 5 concludes with an evaluation of the results and potential implications for future research.

2 MATERIAL AND METHOD

2.1 Dataset

To evaluate the proposed approach, the stroke prediction dataset [24] was used in the study. The dataset consists of 43 400 samples with eleven features. Data with stroke patients in the dataset includes 1.89 % of the entire dataset. This dataset has a typical unbalanced structure, as is often seen in other datasets used in disease detection. The characteristics and values of the data in the dataset are shown in Table 1.

In the stroke prediction dataset, some features such as smoking status and Body Mass Index (BMI) contain missing data, with smoking status having a missing data rate of 30 % and BMI having a rate of 3 %. To prevent these missing data from causing learning problems, used the MissForest algorithm, a Random Forest-based imputation method. The MissForest algorithm identifies attributes with missing data, then assigns the missing values by taking mean-median mode or by random sampling. Additionally, an extra point is created to detect null data. Next, a Random Forest model is created to estimate the missing data. After successful model training, the output is predicted to test the data, and the output is replaced with the relevant data points. This process is repeated until the desired proximity is achieved.

Attributes	**Values**
Patient ID	1–43 400
Gender (gen)	Male/Female
Type of Residence	Rural/Urban
Avg-glucose (glu)	55–291
Work Type (work)	Never-worked/Government Job/ Children/Self-Employed/Private
Smoking Status	Smoked/Formerly/Never
Hypertension (hyp)	Yes/No
Married (mar)	Yes/No
Age	0.08–82
Heart Disease (htd)	Yes/No
Body mass index (BMI)	10.1–97.6

Table 1. Dataset description

2.2 Artificial Immune System

In vertebrates, the immune system consists of B and T lymphocytes that work together to eliminate harmful antigens. When these cells encounter a threat, B lymphocytes produce antibodies that neutralize harmful agents. The genes encoding these antibodies can undergo somatic hypermutation, enhancing the immune system's ability to recognize and combat future pathogens. B cells producing higher-affinity antibodies proliferate more than those producing lower-affinity ones, which strengthens the immune response [25]. Artificial immune systems (AIS) are meta-heuristic methods inspired by the function of B and T lymphocytes in the biological immune system. In AIS, the negative selection algorithm identifies harmful external agents, while ClonALG is responsible for generating antibodies and applying mutations to improve the response to these harmful factors. The overall working mechanism of ClonALG is shown in Figure 1. The ClonALG algorithm incorporates key evolutionary principles: differentiation, diversification, and natural selection. It relies on affinity maturation through hypermutation and selects clones based on their affinity to antigens. Initially, an antibody population is created to represent the immune system, while antigens represent the external agents to be identified. Each antibody is evaluated using the AIS fitness function and calculated separately. The calculated values are copied in proportion to their fitness values, and the created copies are mutated to increase antigen recognition. The generated mutations are then used based on their fitness values, and those not meeting the appropriate criteria are randomly replaced with other clones.

The proposed balancing mechanism in this study employs an Artificial Immune System (AIS) based on the clonal selection theory to enhance the quality of samples within the dataset. ClonALG algorithm prunes unstimulated antibodies and preserves a particular memory by selecting and cloning the most stimulated antibodies. Random mutations occur after cloning, and then affinity calculations are

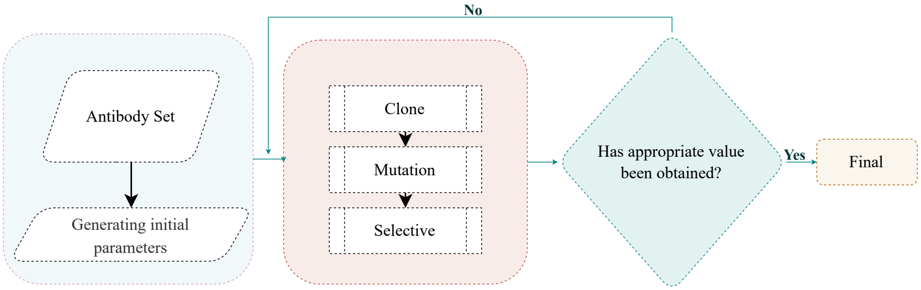
No
Antibody Set
Generating initial parameters
Clone
Mutation
Selective
Has appropriate value been obtained?
Yes
Final

Figure 1. General diagram of the artificial immune system [26]

performed through hypermutation mechanisms, followed by selection based on their antigenic affinity. The mechanism of hypermutation is directly related to the distances between antigens and antibodies, and it is proportional to the affinities of the antibodies. Clones with affinity values below a certain threshold are deleted, while those with appropriate values are retained to ensure the quality of the generated samples [27].

2.3 Firefly Algorithm

The Firefly Algorithm (FFA) is an optimization algorithm inspired by the flashing light behavior of fireflies. Fireflies, which are not distinguished by gender, interact by emitting light at varying brightness levels. These lights attract other fireflies based on their distance and brightness. The FFA mimics this behavior, with each firefly representing a point in the search space. The attractiveness of a firefly is proportional to its objective function value, and fireflies move toward brighter (more attractive) neighbors. This movement allows the algorithm to explore the search space and find near-optimal solutions [28].

In this study, the Firefly Algorithm is employed to enhance the selection step of the Artificial Immune System (AIS) by updating within specific boundaries. This approach ensures the generation of more effective and appropriate data clones, thus improving the overall quality of the solutions. The algorithm uses an objective function where the maximum iteration value (T_{max}) and the lower limit value (L_{min}) are randomly updated within certain bounds during each iteration. This ensures that the cloning process, governed by the artificial immune system, produces more effective results. The objective function used in the algorithm is defined as shown in Equation (1).

$$\text{Objective Function} = \sum_{i=1}^{n} x_i^2. \tag{1}$$

The variable n denotes the dimension of the function to be evaluated, with x_i representing each parameter within the lower and upper limit bounds of L_{min} and

L_{max}, respectively. In the Firefly Algorithm, the most attractive firefly is first determined through the selection of random positions and parameters, which are then evaluated based on the chosen objective function. The objective function selection may vary depending on the specific problem being studied. To provide clarity on the resulting positions, the function outputs are passed through the parameter assignment equation denoted by Equation (2).

$$X_{i,k} = L_{min,k} + \text{rand}(0,1) \times (L_{max,k} - L_{min,k}). \tag{2}$$

In Equation (2), the index i represents the solution set, and k represents the index of the selected parameter within that solution. The value $X_{i,k}$, which denotes the selected parameter for solution i at index k, is generated as a random value between the lower bound $L_{min,k}$ and the upper bound $L_{max,k}$. After determining these parameters, the distance between the calculated values is computed using Equation (3).

$$r_{ij} = \sqrt{\sum_{k=1}^{d} (X_{i,k} - X_{j,k})^2}. \tag{3}$$

r_{ij} is the distance between the i^{th} and j^{th} fireflies. d is the parameter size, and k is the parameter index. The attractiveness between fireflies, denoted as $B(r)$, is then calculated using Equation (4), where γ is the light absorption coefficient, typically set to 1. The attractiveness between fireflies, denoted as $B(r)$, is then calculated using Equation (4), where B_0 represents the base attractiveness or the brightest firefly's attractiveness, and γ is the light absorption coefficient, typically set to 1.

$$B(r) = B_0 e^{-\gamma r^2}. \tag{4}$$

The objective function in this study allows the most attractive features to be maintained within a specific region based on the FFA cycle. Unlike previous methods in the literature, during the selection step of the AIS algorithm, a new selection mechanism is introduced to choose between existing data in memory or newly cloned data. Instead of selecting based on a fixed value, the selection is made according to the FFA attractiveness function, which randomly changes within set limits during each cycle. The experimental results demonstrate that this approach increases the number and diversity of clones produced by the AIS algorithm, as shown in Section 4.

2.4 Artificial Bee Colony Algorithm

The Artificial Bee Colony (ABC) is a meta-heuristic algorithm that imitates the foraging and resource-hoarding behavior of bees. In the foraging process, food sources consist of employed foragers and unemployed gatherers, among other components [24]. Bees perform their food-finding behavior in an organized manner, with worker bees collecting food sources in the hive, onlooker bees monitoring the quality of the food source, and scout bees searching for new food locations randomly. When a scout bee finds a food source, it transforms into a worker bee and takes the nutrients from the sources to the hive, performing various dances during this process and communicating information about the source to the onlooker bee. Based on the information sent by the worker bees, the onlooker bees determine the best-quality food source.

In general, the ABC algorithm works by generating a set of S food source locations, where S represents the size of the worker bees or their food sources. Each nutrient source is defined as $a_i = 1, 2, \dots, S$ and is a vector with dimension D, which represents the number of parameters for the optimization problem. Initially, the food source locations are randomly generated, as in Equation (5).

$$x_{a_i}^{j} = x_{\min}^{j} + \mathrm{rand}(0,1) \cdot \left(x_{\max}^{j} - x_{\min}^{j}\right). \tag{5}$$

The lower and upper values of the j^{th} parameter of the problem, where $j = 1, 2, \dots, D$, are represented by $x_{\max}^{j}$ and $x_{\min}^{j}$, respectively. A random value between 0 and 1 is generated for each of them. Once the location is determined, the fitness values (nectar amounts) of the food sources are evaluated. For the attendant bee, the calculation of the neighboring food source location is shown in Equation (6).

$$V_{a_i}^{j_{\text{rand}}} = x_{\min}^{j_{\text{rand}}} + \mathrm{rand} \cdot \left(x_{a_i}^{j_{\text{rand}}} - x_{a_k}^{j_{\text{rand}}}\right). \tag{6}$$

$x_{a_k}^{j_{\text{rand}}}$ is a randomly selected food source. $a_k \in \{1, 2, \dots, S\}$ is a random integer, and rand is a random value between -1 and 1. D is the number of parameters of the problem at hand.

$$p_{a_i} = \frac{\text{fitness}_{a_i}}{\sum_{n=1}^{S} \text{fitness}_{a_n}}. \tag{7}$$

The fitness value probability is denoted by p_{a_i}, where fitness_{a_i} represents the quality of the food source evaluated by the attendant bee. After calculating the probability, the onlooker bee finds the neighboring food source using Equation (6) and evaluates the nectar amount of the new candidate food source. If the nectar is higher than before, the bee holds the new source position and forgets the old one. A counter is used for each bee working on the food source, and when the counter value exceeds the maximum limit, the bee leaves the food source and looks for new food sources. In this study, the algorithm assigned nutrient sources to each attribute, and the fitness value in Equation (7) was recorded during the iteration. The recorded top seven attributes were used for training. This feature selection process was applied to each randomly generated training dataset during model creation. The cloning and selection process is explained in further detail in Section 3. The experimental results show that this approach increases the number and diversity of clones produced by the AIS algorithm, as discussed in Section 4.

3 PROPOSED MODEL

The general operation of the study involves data acquisition, preprocessing, separation of data for testing and training, application of balancing algorithms, feature selection, training, and classification steps. Missing data are imputed using the MissForest method. After data completion, the data are scaled using the standard scalar method. In the balancing stage, a balanced dataset is created using the one-sided selection algorithm. The performance results of the balanced data are then trained with classifiers in two different ways: using the 7 features determined according to the ABC algorithm and using all features. The general operation of the model is illustrated in Figure 2.

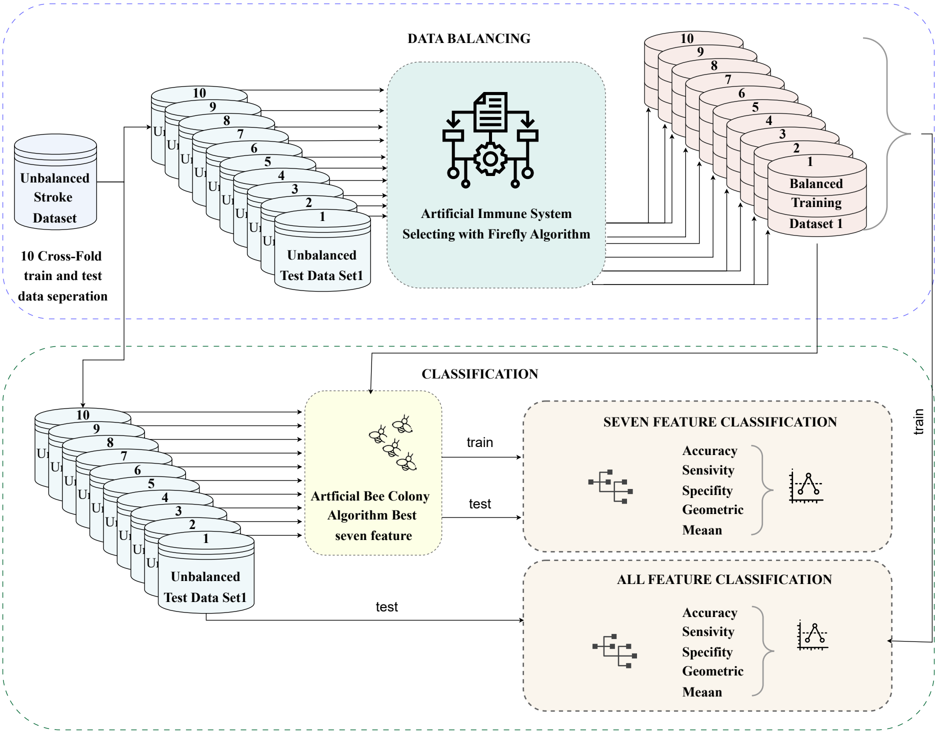
DATA BALANCING
Unbalanced Stroke Dataset
10 Cross-Fold train and test data seperation
10
9
8
7
6
5
4
3
2
1
Unbalanced Test Data Set1
Artificial Immune System Selecting with Firefly Algorithm
Balanced Training Dataset 1
CLASSIFICATION
Artficial Bee Colony Algorithm Best seven feature
train
test
SEVEN FEATURE CLASSIFICATION
Accuracy
Sensivity
Specifity
Geometric
Meaan
ALL FEATURE CLASSIFICATION
Accuracy
Sensivity
Specifity
Geometric
Meaan
train

Figure 2. General block diagram of the proposed model

Before the balancing stage, the data is randomly divided into 10 different sets for training and testing using the 10-fold cross-validation method. The balancing procedure is applied independently to each of the 10 learning datasets, as illustrated in Figure 3. The training datasets are balanced and subsequently used for model training, and their accuracy is evaluated using the corresponding test dataset. After randomly dividing the dataset into 10 different training sets, determined the training parameters for the minority (stroke patient) and majority (non-stroke patient) classes. Antibodies were selected at 20 % for both classes to be used in the artificial immune system. First, the minority and majority datasets were processed separately with their antibodies and antigens in the hybrid ClonALG algorithm to generate new clone data. Then, the clone data generated for the minority and majority classes were combined to create a single dataset. This dataset was processed again in the hybrid ClonALG algorithm with the reserved training data, resulting in new clone data. In the final stage, the majority class data in the clone data and the minority class data generated in the previous steps were combined to balance the dataset.

To achieve successful training results in the unstable stroke dataset, a large number of efficient stroke patient data is required, and the ClonALG algorithm is preferred for this purpose as it enables the production of various clones. FFA was employed in the selection step to increase the number and quality of clones generated by the algorithm. Algorithm 1 shows the part where FFA is incorporated into the artificial immune system. Based on the fitness values generated according to the FFA sphere function, it retains the highest affinity value (q) in the antibody population at the minimum firefly affinity (f). If the highest affinity (p) in the cloned antibodies is less than f, the clone data is replaced with the memory. This step maintains the affinity value within a certain global domain limit and ensures consistent clones are generated. The FFA algorithm parameters used in the study are presented in Table 2. The *lb* and maximum iteration values are randomly determined to prevent the same attractiveness value from occurring each time and to improve the diversity of the produced clones.

Parameter	**Values**
Dimension value	2
Lower value (lb)	random $[1.0, 1.1]$
Upper value (ub)	2
Maximum number of iterations	random $[280, 330]$

Table 2. Firefly algorithm parameter values

Different output values are generated by the FFA algorithm in each iteration. Figure 4 displays the best FFA values obtained by setting the lower limit values as 1.0 and 1.1. The values produced results in the range of 2.0–3.78. The selection step is performed by keeping the algorithm values within a certain limit, which helps in generating consistent results in the produced clones.

Input parameters in the first step of the proposed algorithm: The growth factor of the clone size is determined as β, the size of the population selected for cloning n, and the number of iterations i. The determined antigen set is sent to the algorithm one by one. The affinity values of each identified antibody are calculated according

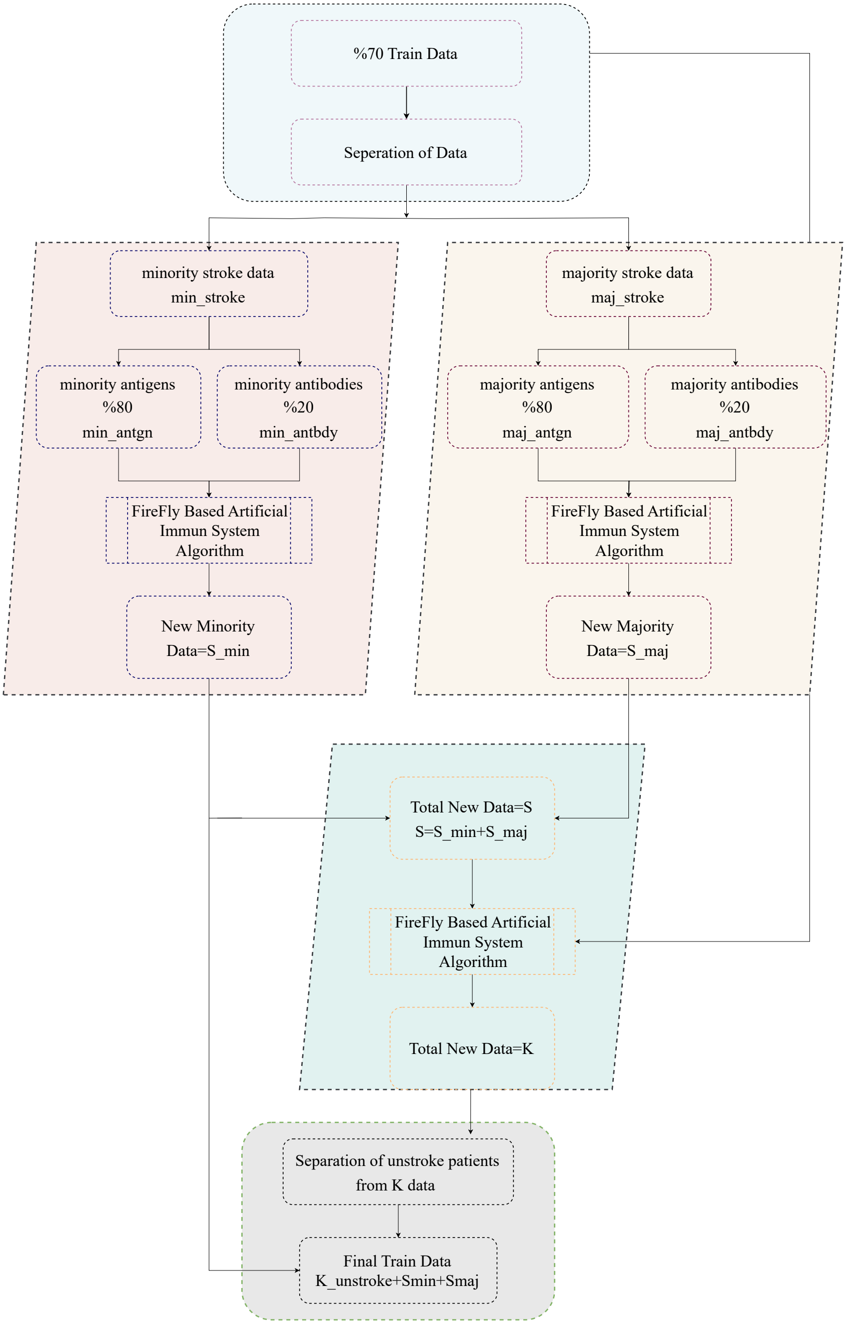

Figure 3. Proposed data balancing algorithm flow chart

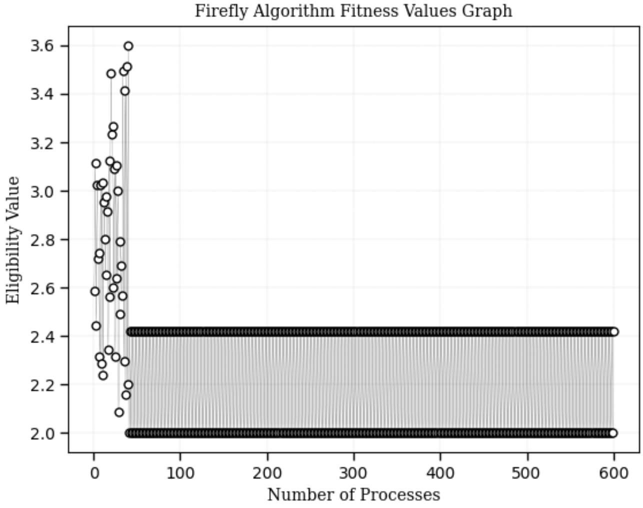

Figure 4. Representation of the best values obtained according to the proposed FFA algorithm

to the antigen. The calculated values are put in order. The first n antibodies sequenced are taken, and for each antibody:

$$B(r) = B_0 e^{-\gamma r^2}. \tag{8}$$

The selected antibody size represented by n and the cloning amplification factor β values result in the production of clones corresponding to the iteration number x associated with the loop. Subsequently, the clones undergo mutation through the Gaussian mutation function. The mutated clone set is denoted as *km*. The clone with the highest affinity in the *km* set is denoted as p, while the antibody with the highest affinity in the data set is denoted as q. Following the determination of the most suitable value f through the firefly algorithm, clones with affinities exceeding f are disregarded. Thus, for each iteration, clones with affinities other than the firefly's best fitness value are neglected. The pseudo-code of the algorithm is presented in Algorithm 1. The initial fixed variables β, n, and G are used, where β represents the cloning growth factor $\beta = 1$, n represents the size of the selected population for cloning $n = 10$, and G represents the number of generations $G = 1$ for *Smin* and *Smaj* clone production, respectively.

The clones generated by the proposed hybrid model were arranged according to the OSS algorithm to balance the minority and majority classes. Firstly, the antigens and antibodies belonging to the minority class and those belonging to the majority class were passed separately through the algorithm proposed in Algorithm 1. The stroke patient data in the productive clones obtained from both algorithms were extracted and added to the balanced dataset created for the final stage. Then,

Algorithm 1 Cloning and Mutation Algorithm

```
Require: $\beta$, $n$, $N$, $G$
while the length of antigen data is not reached do
    for $j = 1$ to $N$ do
        for $g = 1$ to $G$ do
            Calculate affinity for each antibody
            Add to affinity list
            Sort the list
        end for
        for $i = 1$ to $n$ do
            $x \leftarrow (\beta \cdot N)/i$
            Produce $x$ number of clones
            Mutate the clones (km)
            $p \leftarrow$ antibody with the highest affinity in km
            $q \leftarrow$ antibody with the highest affinity in the dataset
            $f \leftarrow$ optimal value determined by firefly algorithm
        end for
        if $f > q$ then
            $f \leftarrow q$
        end if
        if $f > p$ then
            Replace data in memory with cloned data
        end if
    end for
    Append the data from the determined iteration to a .csv file
end while
```

all outputs obtained by crossing the hybrid model with the minority and majority class's antigens and antibodies were combined. This new dataset, along with the entire learning dataset, was used in the hybrid model to generate non-patient data. While all the data allocated for training were set as antigens, the combined dataset with the previously formed clones became antibodies. In this step, the data without stroke patients were also added to the balanced dataset. As a result of these steps, the patient data rate increased from 1.8 % to at least 15.5 % for each training dataset created in the cross-fold. Following the completion of the balancing procedures, the proposed model was classified into two stages. In the first stage, all features were evaluated by using. In the second stage, the feature selection process was performed by the Artificial Bee Colony (ABC) algorithm. The performance of the algorithm is determined by essential parameters such as the number of features (D), colony size (S), maximum iteration number, and trial limit. The number of features $D = 7$ determines the size of each food source in the optimization process (as shown by Equation (5)). The colony size $S = 20$, i.e., the number of food sources, was determined as 20. This means that worker bees will explore 20 different subsets of features in each iteration. The algorithm was run for a maximum of 50 iterations to perform the search efficiently. In each iteration, food sources are trained with the selected features according to their fitness values, and the results are recorded. The fitness values determine the quality of the feature sets, and for this stage, we use Equation (7). In order to maintain the diversity of the population, if a food source does not show improvement in training results after 20 consecutive trials, it is randomly replaced with a new source. This mechanism prevents the algorithm from getting stuck in local minima. This process was applied separately for the selected dataset in each cross-fold. Each of them was trained according to its own best attribute. This resulted in better performance compared to using all components in training. During the concluding phase, the clusters created by the proposed model were evaluated using six machine learning algorithms.

The classifiers, CatBoost, Gradient Boosting, XGBoost, LightGBM, Logistic Regression, and Support Vector Machine (SVM), are models that have proven to deliver high-performance results for various data types [29, 30, 31]. Each classifier has its own unique characteristics that distinguish it from the others. CatBoost is typically preferred for categorical data, Gradient Boosting excels in model optimization, XGBoost is highly effective for large datasets due to its speed and accuracy, and LightGBM, with its simple structure, also achieves high accuracy on large datasets [32, 33]. Logistic Regression performs well in binary classification problems, while SVM, which separates classes using the maximum margin principle, is effective in complex classification tasks [34, 35]. The reason for selecting these classifiers is that they offer flexibility, accuracy, speed, and overall performance advantages for different data types, allowing a comprehensive approach to the dataset.

For the imbalanced dataset, GridSearchCV, a method proven to produce effective results in various classification applications by optimizing model hyperparameters, was used to achieve high sensitivity and specificity, especially on the test set [36, 37]. GridSearchCV is a systematic hyperparameter optimization method in which a specified range of values for each parameter is exhaustively searched to find the combination that maximizes model performance [38]. An algorithm is developed that automatically selects the optimal hyperparameter configuration and optimizes until it reaches at least 70 % specificity and sensitivity on the test data. In this algorithm, parameters such as tree depth (the maximum depth of trees in the model), learning rate (which controls the step size at each iteration of model training), and iterations (the number of boosting rounds or trees) are tuned for decision tree-based algorithms like CatBoost, XGBoost, and LightGBM. For instance, for the CatBoost algorithm, tree depth is selected between 6, 8, and 10, learning rate between 0.01, 0.05, and 0.1, and iterations between 100 and 200. For XGBoost, tree depth is set between 3, 6, and 9, learning rate between 0.01, 0.05, and 0.1, and the number of trees (*n_estimators*) is between 100 and 200. Similarly, in LightGBM, tree depth is adjusted between -1, 6, and 8, learning rate between 0.01, 0.05, and 0.1, and the number of trees between 100 and 200. Additionally, for Logistic Regression, the regularization parameter C is chosen between 0.01, 0.1, 1, and 10, with solver options 'liblinear' and 'lbfgs' being optimized (liblinear is suitable for small datasets and linear models, whereas lbfgs is a more appropriate optimization algorithm for larger datasets). For SVM, the regularization parameter C is selected between 0.1, 1, and 10, with kernel functions 'linear' and 'rbf' being utilized ('linear' uses a linear decision boundary, while 'rbf' kernel creates a more complex decision boundary for non-linear data). By tuning the parameters in this way, the best possible configuration of the models is selected, which leads to more reliable and accurate predictions. And the performance metrics are constructed comparatively. All the outputs of the proposed model are presented in Section 4.

4 EXPERIMENTAL RESULTS

In the study, the performance of the balanced dataset is evaluated with six different learning algorithms, namely, newly developed algorithms such as CatBoost, LightGBMBoost, GB, XGBoost, SVM, and LR algorithms, as shown in Table 3. The evaluation criteria used to prove the effectiveness of these classification algorithms in the proposed model are, respectively, accuracy (Acc.), sensitivity (Sen.), specificity (Spec.), and geometric mean (G-Mean). In the first step, when calculating performance metrics to solve the class imbalance problem, the four metrics based on the prediction results create a confusion matrix. These metrics are true positive (TP), false negative (FN), false positive (FP), and true negative (TN) [39, 40].

The accuracy value (Acc.) expresses the accuracy rate of the prediction made by the model.

$$\text{Accuracy (Acc.)} = \frac{TP + TN}{TP + FP + TN + FN}. \tag{9}$$

Sensitivity is the measure of each category's ability to predict true positives. It refers to the rate of correctly guessing that the person is sick.

$$\text{Sensitivity (Sens.)} = \frac{TP}{TP + FN}. \tag{10}$$

Specificity helps predict the true negatives of each category in making. It refers to the correct guess that the person is not sick.

$$\text{Specificity (Spec.)} = \frac{TN}{TN + FP}. \tag{11}$$

The geometric mean (G-Mean) is used to evaluate the degree of equilibrium of the algorithm, especially for unbalanced class data, and the larger the value, the better the result.

$$\text{G-mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}}. \tag{12}$$

SVM, which is one of the classifiers used to evaluate performance metrics, provides nonlinear classification by determining the optimum plane between the planes in the data and can organize the data in high dimensions. They can be used as effective classifiers in studies in different fields [29]. In the proposed study with the best seven selections, higher results were produced with SVM in two performance metrics, with 82 % sensitivity and 87 % G-Mean. The LR algorithm, which statistically determines the relationship between the variables, is not much affected by the connection complexity and noise in the data. Due to this advantage, it has been one of the preferred methods in this study, and comparable consistent results have been produced [35].

Boosting algorithms are generally used to strengthen weak learning algorithms and produce strong results from weak data, thanks to the weights assigned in each iteration with the iterative learning method. The algorithms used in this study are Gradient Boosting, Extreme Gradient Boosting (XGBoost), LightGBMBoost, and CatBoost. Gradient Boosting is an algorithm that advances in training by calculating the previous error and adding the previous predictions while creating new predictions. XGBoost was developed with Gradient Boosting and includes improvements such as pruning, tolerating missing values, and removing deviations. Additionally, it is better than Gradient Boosting in terms of avoiding overfitting and improving training speed [32]. LGBMBoost increases training speed by offering features such as Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) and provides convenience in calculations by using variables discretely [35]. CatBoost, on the other hand, is more effective than other boosting algorithms in processing data with different structures faster [30].

In the experiment where all features were used, GradientBoost gave the best accuracy with 94 % and specificity with 94 % values, SVM gave the best sensitivity with 80 % and G-Mean with 86 % values. When the result table is evaluated, the SVM algorithm produces the most appropriate values since it has the highest sensitivity and G-Mean values.

Classification	**Acc.**	**Spec.**	**Sens.**	**G-Mean**
CatBoost	0.939	0.942	0.769	0.850
GradientBoost	**0.942**	**0.945**	0.738	0.834
XGBoost	0.941	0.944	0.728	0.829
LightGBMBoost	0.939	0.943	0.716	0.821
LR	0.933	0.936	0.772	0.850
SVM	0.929	0.931	**0.803**	**0.865**

Table 3. Comparison of performance metrics with all attributes

In the experiment where the seven best features were used, GradientBoost gave the best accuracy with 94 % and specificity with 93 % values, SVM achieved the highest sensitivity with 82 % value, and the best G-Mean with 87 % value according to the performance results obtained with the enhancement algorithms. When the result Table 4 was evaluated, it was seen that the SVM algorithm provided the most consistent results with 93 % specificity, 87 % geometric mean, 93 % accuracy, and 82 % sensitivity in correctly predicting non-patients as healthy and sick individuals as sick.

Classification	**Acc.**	**Spec.**	**Sens.**	**G-Mean**
CatBoost	0.937	0.939	0.778	0.854
GradientBoost	**0.942**	**0.946**	0.728	0.829
XGBoost	0.939	0.943	0.719	0.823
LightGBMBoost	0.939	0.942	0.734	0.831
LR	0.933	0.936	0.772	0.850
SVM	0.928	0.930	**0.816**	**0.871**

Table 4. Comparison of performance metrics with best attributes

For addressing the stroke data balancing problem, Liu et al. [15] performed class estimation with a real-time classifier with auto-selected hyperparameters using the same dataset as this study. They solved the missing data problem in the data using the RF algorithm, and in the data balancing step, using PCA and K-means clustering methods. In this study, a geometric mean of 47 %, a sensitivity of 67 %, a specificity of 32 %, and an accuracy of 71 % were produced. Building upon the work of Liu et al. [15], Santos et al. [16] addressed the missing data in the preprocessing step by deleting it. Additionally, they balanced the data using hyperparameters in the OSS model with the artificial immunity algorithm updated according to the Kohonen network. Then, using decision trees induced by genetic programming and arranged in an interpretable structure, they achieved 74 % geometric mean, 78 % sensitivity, 70 % specificity, and 70 % accuracy results. Liu et al. [15] presented their results comparatively. It is shown in Table 5 that the proposed model produced results comparable to these previous studies in the literature. In addition, the unbalanced stroke dataset named ”Raw Data” was trained directly with the proposed classification model, and the results were presented. Although the proposed model exhibits very high accuracy (98 %) and specificity (99 %) for the unbalanced dataset, the sensitivity rate (0.01 %), which reflects the model's ability to correctly identify stroke patients, is extremely low. This indicates that the learning approach is not suitable for this dataset.

This study has achieved higher performance outcomes compared to the work of Liu et al. [15] and the study by Santos et al. [16]. The success of the applied model has been demonstrated through all performance metrics, providing strong evidence of its effectiveness.

5 CONCLUSIONS

This study aimed to increase the reliability of stroke diagnosis based on patients' physiological characteristics while considering the data imbalance problem. A stroke data estimation model balanced according to the OSS model was proposed to achieve this goal using the AIS algorithm selected with FFA. The study provided consistent results comparable to many performance metrics for the imbalance problem in both patient and non-patient classes. It has been demonstrated that the proposed model's precision and accuracy are more effective than those of the two models proposed

Study	Pre Process	Data Balancing	Feature Selection	Classi-fication	Acc.	Spec.	Sens.	G-Mean
[16]	Delete missing data	Kohonen selective AIS, OSS	–	DT-GP	0.70	0.70	0.78	0.74
[15]	Random Forest	PCA, K-Mean	–	AutoHPO	0.71	0.32	0.67	0.47
Raw Data	Miss Forest	–	ABC Algorithm	GBoost	**0.98**	**0.99**	0.01	0.09
Prop. 1	Miss Forest	FFA selective AIS, OSS	–	SVM	0.93	0.93	0.80	0.86
Prop. 2	Miss Forest	FFA selective AIS, OSS	ABC Algorithm	SVM	0.93	0.93	**0.82**	**0.87**

Table 5. Literature studies and proposed model comparison

in previous studies in the literature. In future studies, the same model can be attempted in other datasets with an unstable database, apart from the diagnosis of stroke disease. Additionally, by determining higher dimensional selection thresholds in the FFA algorithm, a more homogeneous and diverse production of resulting clones can be achieved, which may improve educational performance.

REFERENCES

[1] Owolabi, M. O.—Thrift, A. G.—Martins, S.—Johnson, W.—Pandian, J. et al.: The State of Stroke Services Across the Globe: Report of World Stroke Organization – World Health Organization Surveys. International Journal of Stroke, Vol. 16, 2021, No. 8, pp. 889–901, doi: 10.1177/17474930211019568.

[2] Capirossi, C.—Laiso, A.—Renieri, L.—Capasso, F.—Limbucci, N.: Epidemiology, Organization, Diagnosis and Treatment of Acute Ischemic Stroke. European Journal of Radiology Open, Vol. 11, 2023, Art. No. 100527, doi: 10.1016/j.ejro.2023.100527.

[3] Chen, Y.—Abel, K. T.—Janecek, J. T.—Chen, Y.—Zheng, K.—Cramer, S. C.: Home-Based Technologies for Stroke Rehabilitation: A Systematic Review. International Journal of Medical Informatics, Vol. 123, 2019, pp. 11–22, doi: 10.1016/j.ijmedinf.2018.12.001.

[4] O'Donnell, M. J.—Chin, S. L.—Rangarajan, S.—Xavier, D.—Liu, L. et al.: Global and Regional Effects of Potentially Modifiable Risk Factors Associated with Acute Stroke in 32 Countries (INTERSTROKE): A Case-Control Study. The Lancet, Vol. 388, 2016, No. 10046, pp. 761–775, doi: 10.1016/S0140-6736(16)30506-2.

[5] Arslan, A. K.—Colak, C.—Sarihan, M. E.: Different Medical Data Mining Approaches Based Prediction of Ischemic Stroke. Computer Methods and Programs in Biomedicine, Vol. 130, 2016, pp. 87–92, doi: 10.1016/j.cmpb.2016.03.022.

[6] Puspitasari, D. I.—Riza Kholdani, A. F.—Dharmawati, A.—Rosadi, M. E.—Dhuhita, W. M. P.: Stroke Disease Analysis and Classification Using Decision Tree and Random Forest Methods. 2021 Sixth International Conference on Informatics and Computing (ICIC), 2021, pp. 1–4, doi: 10.1109/ICIC54025.2021.9632906.

[7] Haixiang, G.—Yijing, L.—Shang, J.—Mingyun, G.—Yuanyue, H.—Bing, G.: Learning from Class-Imbalanced Data: Review of Methods and Applications. Expert Systems with Applications, Vol. 73, 2017, pp. 220–239, doi: 10.1016/j.eswa.2016.12.035.

[8] Liz, H.—Huertas-Tato, J.—Sánchez-Montañés, M.—Del Ser, J.—Camacho, D.: Deep Learning for Understanding Multilabel Imbalanced Chest X-Ray Datasets. Future Generation Computer Systems, Vol. 144, 2023, pp. 291–306, doi: 10.1016/j.future.2023.03.005.

[9] Bhati, A.—Gour, N.—Khanna, P.—Ojha, A.: Discriminative Kernel Convolution Network for Multi-Label Ophthalmic Disease Detection on Imbalanced Fundus Image Dataset. Computers in Biology and Medicine, Vol. 153, 2023, Art. No. 106519, doi: 10.1016/j.compbiomed.2022.106519.

[10] Li, J.—Liu, L. S.—Fong, S.—Wong, R. K.—Mohammed, S.—Fiaidhi, J.—Sung, Y.—Wong, K. K. L.: Adaptive Swarm Balancing Algorithms for Rare-Event Prediction in Imbalanced Healthcare Data. PLoS ONE, Vol. 12, 2017, No. 7, Art. No. e0180830, doi: 10.1371/journal.pone.0180830.

[11] Yagin, F. H.—Cicek, I. B.—Kucukakcali, Z.: Classification of Stroke with Gradient Boosting Tree Using SMOTE-Based Oversampling Method. Medicine Science – International Medical Journal, Vol. 10, 2021, No. 4, pp. 1510–1515, doi: 10.5455/medscience.2021.09.322.

[12] Sailasya, G.—Kumari, G. L. A.: Analyzing the Performance of Stroke Prediction Using ML Classification Algorithms. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 12, 2021, No. 6, pp. 539–545, doi: 10.14569/ijacsa.2021.0120662.

[13] Rana, C.—Chitre, N.—Poyekar, B.—Bide, P.: Stroke Prediction Using Smote-Tomek and Neural Network. 2021 12$^{\text{th}}$ International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, pp. 1–5, doi: 10.1109/ICCCNT51525.2021.9579763.

[14] Dev, A.—Malik, S. K.: Artificial Bee Colony Optimized Deep Neural Network Model for Handling Imbalanced Stroke Data: ABC-DNN for Prediction of Stroke. International Journal of E-Health and Medical Communications, Vol. 12, 2021, No. 5, pp. 67–83, doi: 10.4018/ijehmc.20210901.oa5.

[15] Liu, T.—Fan, W.—Wu, C.: A Hybrid Machine Learning Approach to Cerebral Stroke Prediction Based on Imbalanced Medical Dataset. Artificial Intelligence in Medicine, Vol. 101, 2019, Art. No. 101723, doi: 10.1016/j.artmed.2019.101723.

[16] Santos, L. I.—Camargos, M. O.—D'Angelo, M. F. S. V.—Mendes, J. B.—

De Medeiros, E. E. C.—Guimarães, A. L. S.—Palhares, R. M.: Decision Tree and Artificial Immune Systems for Stroke Prediction in Imbalanced Data. Expert Systems with Applications, Vol. 191, 2022, Art. No. 116221, doi: 10.1016/j.eswa.2021.116221.

[17] Hassan, S. M.—Ali, S. A.—Hassan, B.—Hussain, I.—Rafiq, M.—Awan, S. A.: Hybrid Features Binary Classification of Imbalance Stroke Patients Using Different Machine Learning Algorithms. International Journal of Biology and Biomedical Engineering, Vol. 16, 2022, pp. 154–160, doi: 10.46300/91011.2022.16.20.

[18] Zhang, Y.—Zhu, D.—Li, T.—Wang, X.—Zhao, L.: Detection of Acute Ischemic Stroke and Backtracking Stroke Onset Time via Machine Learning Analysis of Metabolomics. Biomedicine Pharmacotherapy, Vol. 155, 2022, Art. No. 113641, doi: 10.1016/j.biopha.2022.113641.

[19] Biswas, N.—Uddin, K. M. M.—Rikta, S. T.—Dey, S. K.: A Comparative Analysis of Machine Learning Classifiers for Stroke Prediction: A Predictive Analytics Approach. Healthcare Analytics, Vol. 2, 2022, Art. No. 100116, doi: 10.1016/j.health.2022.100116.

[20] Srinivas, A.—Mosiganti, J. P.: A Brain Stroke Detection Model Using Soft Voting Based Ensemble Machine Learning Classifier. Measurement: Sensors, Vol. 29, 2023, Art. No. 100871, doi: 10.1016/j.measen.2023.100871.

[21] Ahammad, T.: Risk Factors Identification for Stroke Prognosis Using Machine Learning Algorithms. Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 8, 2022, No. 3, pp. 282–296, doi: 10.5455/jjcit.71-1652725746.

[22] Sun, Y.—Li, J.—Xu, Y.—Zhang, T.—Wang, X.: Deep Learning Versus Conventional Methods for Missing Data Imputation: A Review and Comparative Study. Expert Systems with Applications, Vol. 227, 2023, Art. No. 120201, doi: 10.1016/j.eswa.2023.120201.

[23] Cooper, E. L.: Evolution of Immune Systems from Self/Not Self to Danger to Artificial Immune Systems (AIS). Physics of Life Reviews, Vol. 7, 2010, No. 1, pp. 55–78, doi: 10.1016/j.plrev.2009.12.001.

[24] Karaboga, D.: An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report. Erciyes University, Engineering Faculty, Computer Engineering Department, 2005, `https://abc.erciyes.edu.tr/pub/tr06_2005.pdf`.

[25] Timmis, J.—Hone, A.—Stibor, T.—Clark, E.: Theoretical Advances in Artificial Immune Systems. Theoretical Computer Science, Vol. 403, 2008, No. 1, pp. 11–32, doi: 10.1016/j.tcs.2008.02.011.

[26] Liu, T.—Fan, W.—Wu, C.: A Hybrid Machine Learning Approach to Cerebral Stroke Prediction Based on the Imbalanced Medical Dataset. Artificial Intelligence in Medicine, Vol. 101, 2019, Art. No. 101723, doi: 10.1016/j.artmed.2019.101723.

[27] Fister Jr, I.—Yang, X. S.—Fister, I.—Brest, J.: Memetic Firefly Algorithm for Combinatorial Optimization. CoRR, 2012, doi: 10.48550/arXiv.1204.5165 (Accessed: Feb. 19, 2023).

[28] Chawla, N. V.: Data Mining for Imbalanced Datasets: An Overview. In: Maimon, O., Rokach, L. (Eds.): Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA, 2009, pp. 875–886, doi: 10.1007/978-0-387-09823-4_45.

[29] D'Angelo, M. F. S. V.—Palhares, R. M.—Camargos Filho, M. C. O.—Maia, R. D.—Mendes, J. B.—Ekel, P. Y.: A New Fault Classification Approach Applied to Tennessee Eastman Benchmark Process. Applied Soft Computing, Vol. 49, 2016, pp. 676–686, doi: 10.1016/j.asoc.2016.08.040.

[30] Modak, S. K. S.—Jha, V. K.: Diabetes Prediction Model Using Machine Learning Techniques. Multimedia Tools and Applications, Vol. 83, 2024, No. 13, pp. 38523–38549, doi: 10.1007/s11042-023-16745-4.

[31] Zapata-Cortes, O.—Arango-Serna, M. D.—Zapata-Cortes, J. A.—Restrepo-Carmona, J. A.: Machine Learning Models and Applications for Early Detection. Sensors (Basel), Vol. 24, 2024, No. 14, Art. No. 4678, doi: 10.3390/s24144678.

[32] Mahajan, P.—Uddin, S.—Hajati, F.—Moni, M. A.: Ensemble Learning for Disease Prediction: A Review. Healthcare, Vol. 11, 2023, No. 12, Art. No. 1808, doi: 10.3390/healthcare11121808.

[33] Sinha, B. B.—Ahsan, M.—Dhanalakshmi, R.: LightGBM Empowered by Whale Optimization for Thyroid Disease Detection. International Journal of Information Technology, Vol. 15, 2023, No. 4, pp. 2053–2062, doi: 10.1007/s41870-023-01261-3.

[34] Guido, R.—Ferrisi, S.—Lofaro, D.—Conforti, D.: An Overview on the Advancements of Support Vector Machine Models in Healthcare Applications: A Review. Information, Vol. 15, 2024, No. 4, Art. No. 235, doi: 10.3390/info15040235.

[35] Ma, J.—Dhiman, P.—Qi, C.—Bullock, G.—van Smeden, M.—Riley, R. D.—Collins, G. S.: Poor Handling of Continuous Predictors in Clinical Prediction Models Using Logistic Regression: A Systematic Review. Journal of Clinical Epidemiology, Vol. 161, 2023, pp. 140–151, doi: 10.1016/j.jclinepi.2023.07.017.

[36] Sahoo, S.—Bisoy, S. K.—Mallick, P. K.: Improving Autism Detection Using GridSearchCV for Severity Level Autism in Indonesian Children. 2024 7th International Conference on Informatics and Computational Sciences (ICICoS), IEEE, 2024, pp. 468–473, doi: 10.1109/ICICoS62600.2024.10636869.

[37] Verma, B. K.—Yadav, A. K.: Advancing Software Vulnerability Scoring: A Statistical Approach with Machine Learning Techniques and GridSearchCV Parameter Tuning. SN Computer Science, Vol. 5, 2024, No. 5, Art. No. 595, doi: 10.1007/s42979-024-02942-x.

[38] Pedregosa, F.—Varoquaux, G.—Gramfort, A.—Michel, V.—Thirion, B. et al.: Scikit-Learn: Machine Learning in Python. Journal of Machine Learning Research, Vol. 12, 2011, pp. 2825–2830, `http://jmlr.org/papers/v12/pedregosa11a.html`.

[39] Galar, M.—Fernandez, A.—Barrenechea, E.—Bustince, H.—Herrera, F.: A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 42, 2012, No. 4, pp. 463–484, doi: 10.1109/TSMCC.2011.2161285.

[40] Rady, E. H. A.—Anwar, A. S.: Prediction of Kidney Disease Stages Using Data Mining Algorithms. Informatics in Medicine Unlocked, Vol. 15, 2019, Art. No. 100178, doi: 10.1016/j.imu.2019.100178.

Şerife Çelikbaş works in the field of artificial intelligence in biomedical technologies. She completed her undergraduate and graduate studies in biomedical engineering. She continues her doctoral studies in the same field at the Istanbul University-Cerrahpaşa, Biomedical Engineering Department.

Zeynep Orman is an expert in the field of computer science engineering. She received her B.Sc., M.Sc., and Ph.D. degrees in computer engineering from the Istanbul University. She conducted her postdoctoral research at the Brunel University, United Kingdom. Currently, she is working as Professor in the Department of Computer Engineering at the Istanbul University-Cerrahpaşa, Faculty of Engineering.

Abdurrahim Akgündoğdu is an electrical, electronic, and biomedical engineering expert. He completed his undergraduate studies at Istanbul University and subsequently earned his Master's and Ph.D. in biomedical engineering at the same institution. He also pursued doctoral research at the Université d'Orléans in France. Currently, he works as Associate Professor at the Istanbul University-Cerrahpaşa, Department of Electrical and Electronics Engineering.