https://doi.org/ 10.33472/AFJBS.6.4.2024.310-331

African Journal of Biological Sciences

Research Paper Open Access

Biological Data Analysis for Diabetes and Leukemia Detection using Hybrid Classification Approach

M. Sivaraman[1]*

[1]PhD Research Scholar, Department of Computer Science, Dr. SNS Rajalakshmi College of Arts and Science, Coimbatore, Tamil Nadu-641049, India

Email: sivaramanranjith@gmail.com

Dr. J. Sumitha[2]

[2]Assistant Professor, Department of Computer Science, Dr. SNS Rajalakshmi College of Arts and Science, Coimbatore, Tamil Nadu-641049, India

Article History
Volume 6,Issue 4, Feb 2024
Received:17 Feb 2024
Accepted : 01 Mar 2024
doi: 10.33472/AFJBS.6.4.2024.310-331

Abstract: Biological data analysis is an important approach which utilizes genomic, transcriptomics, proteomic, metabolomics, or clinical data for disease detection process. Diabetes and Leukemia are two distinct medical conditions, but research has shown type 2 diabetes patients have a 20% greater risk of being affected by blood cancers, like acute leukemia, indicating the relationship between the two diseases. Early detection of these diseases by analyzing the biological datasets is essential for providing prognostic support. However, the class imbalance and high dimensionality problems in Machine Learning (ML)-based techniques have often degraded effective analysis of clinical and genomic datasets for disease detection. This paper focuses on developing an efficient clinical decision support system using advanced metaheuristic and ML algorithms to solve class imbalance and high dimensionality problems. The first stage of the proposed approach utilizes an optional data augmentation and another pre-processing method for outlier detection and removal using Modified Z-Score (MZS) based on the Median Absolute Deviation (MAD) metric. Then, the optimal features/genes are selected using a hybrid Firefly Pearson's Correlation Coefficient (FPCC)-based Feature/Gene Selection method to reduce the higher feature dimensionality problem. Once the features/genes are selected, the proposed Ladybug Beetle Optimized Universum Learning-based Twin Boosted Adaptive Support Vector Machine (LBO-ULTBASVM) classifier detects the disease with reduced model complexity and error rates. LBO-ULTBASVM is developed by improving the Twin Support Vector Machine (TSVM) classifier by integrating the Universum Learning, Ladybug Beetle Optimization (LBO), and XGBoost for solving the class imbalance problem, reducing training time and improving disease accuracy. Experiments are conducted using PIMA Indians Diabetes and GSE9476 Leukemia datasets and the outcomes indicated that the LBO-ULTBASVM-based model increases the diabetes and leukemia detection accuracy with reduced model complexity and processing time.

Keywords: Biological data analysis, Diabetes, Leukemia, Modified Z-Score, Firefly Pearson's Correlation Coefficient, Ladybug Beetle Optimization, Universum Learning, Twin Boosted Adaptive Support Vector Machine.

1. INTRODUCTION

Biological data analysis for disease detection is a multidisciplinary effort that integrates expertise from biology, bioinformatics, computational biology, statistics, and clinical medicine to improve the understanding of disease mechanisms and develop more effective strategies for diagnosis, treatment, and prevention. Biological data analysis involves the application of various computational and statistical techniques to interpret large-scale biological datasets with the goal of identifying patterns, biomarkers, or signatures associated with specific diseases [1]. Initially, the diverse biological data types, such as genomic, transcriptomics, proteomic, metabolomics, or clinical data are collected. These data may be obtained from patient samples, and cell lines. These raw datasets are cleaned and pre-processed by applying quality control, data imputation, batch correction and denoising methods to remove noise, correct errors, and normalize the data for ensuring consistency. Then, the relevant features from these pre-processed biological data are extracted and the dimensionality is reduced using suitable statistical or feature selection techniques. Finally, the statistical tests or ML algorithms are used to analyze the processed data and identify the patterns that are associated with the disease. The results of the analysis are used to identify potential biomarkers, genetic variants, gene expression signatures, or other molecular features associated with disease presence, progression, or response to treatment. These biomarkers may serve as diagnostic tools, prognostic indicators, or therapeutic targets for the disease. These findings are translated to the clinical practice by developing diagnostic assays, prognostic tests, or personalized treatment strategies based on the identified biomarkers.

In this paper, Diabetes Mellitus (DM), commonly called diabetes, and leukemia, two of the serious medical conditions faced by a considerable population in the modern world, are considered. DM is a common non-communicable chronic metabolic disorder characterized by elevated blood sugar levels over an extended time, resulting from inadequate insulin secretion. Type 1 diabetes mellitus (T1DM) and Type 2 diabetes mellitus (T2DM) are the two common types, with T2DM contributing to almost 90% of all diabetes cases. While T1DM, also called juvenile diabetes, is caused mostly by an autoimmune reaction of the body where the pancreas stops producing insulin, T2DM is caused by inadequate insulin absorption or the body's resistance to insulin. According to the World Health Organization (WHO), 422 million people have diabetes globally in all income countries, and 1.5 million deaths are caused directly by diabetes each year. T2DM has been found to cause diabetic kidney disease, retinopathy, pancreatic failure, joint failures, and immunosuppression, which might increase the risks of other chronic diseases like heart disease and cancers. The terrifying side effects of T2DM also include malfunctioning and permanent damage to body organs [2]. India has recorded 700000 deaths related to direct and indirect complications of diabetes in 2020. Recent studies have shown that T2DM patients have a 20% higher risk of incurring blood cancers such as leukemia, anemia, and thalassemia. Among them, leukemia is another serious health condition in the bone marrow and blood where abnormal white blood

cells (WBC) are produced in large quantities, crowding the normal cells in the immune system due to immunosuppression [3]. Acute lymphocytic Leukemia (ALL), acute myeloid Leukemia (AML), chronic lymphocytic Leukemia (CLL), and chronic myeloid Leukemia (CML) are the four types of leukemia. ALL and AML are common in children and adults, respectively, while CLL and CML are diagnosed commonly in older adults [4]. According to WHO reports, 311594 leukemia deaths have been recorded in 2020, of which 33383 constituting 0.39% of total national deaths have occurred in India. While there is no direct link between diabetes and leukemia, there might be shared risk factors. The diseases have several indirect connections and shared risk factors, such as multifactorial genetic mutations, chronic inflammation, and immunosuppression. Additionally, individuals with diabetes might have a slightly higher risk of certain infections, potentially impacting the immune system and indirectly influencing leukemia risk [5].

Early detection of diabetes and leukemia is commonly performed using clinical decision-making methods that employ advanced data processing approaches. By using gene expression profiles leukemia is learnt accurately and by using the standard clinical data diabetes is learnt. For multiple detection of disease studies have been done by the researchers using DL and ML methods [6]. Although efficient results have been achieved, various factors still negatively impact the real-time analysis for disease detection. The effective analyses have often degraded by class imbalance and high dimensionality problem for disease detection of clinical dataset. In gene expression analysis this is a major issue for classification is to learn many characteristics which creates an impact in accuracy for detection of leukemia [7]. Therefore, it is important to develop a clinical decision support system in advanced to degrade the class imbalance and dimensionality issues and to identify and learn the characteristics of diabetes and leukemia. This paper creates a meta-heuristic method to ensure and enhance the disease detection by considering the limitation and issues as the objective to develop a ML-optimized classifier.

The input dataset are collected from the public repositories. For improving the synthetic class sample, preprocessing step along with data augmentation is executed as the process of outlier detection. For the removal and detection of outlier, MZS parameter along with MAD is utilized. FPCC-based feature/gene selection method is used for selecting and extracting the features after the outlier is removed. By integrating the firefly algorithm and PCC method the hybrid method of FPCC is created as a wrapper method. The diabetes and leukemia dataset ha many features which lead to dimensionality issues, so that the FPCC helps in selecting and extracting only the important features related to the classification for detection of disease. Finally, the proposed LBO-ULTBASVM classifier is used for final classification and disease detection. This LBO-ULTBASVM is a combination of many ML concepts and metaheuristic algorithms. The TSVM classifier is used as the base model, and it is enhanced by including the Universum Learning for class distinction, XGBoost for performance boosting and LBO [8] for parameter tuning. The Adaptive parameter θ is varied to obtain different loss functions for the Twin SVMs. This classifier model is intended to reduce the model complexity and training time by tuning the parameters using LBO, solving the class imbalance problem by Universum learning, improving the classifier accuracy, and

reducing the error rates by embedding XGBoost as a Boosting Technique. Experiments are performed on the above-specified benchmark datasets to assess the performance of the presumed LBO-ULTBASVM-based approach for accurate diabetes and leukemia disease detection. The following sessions are documented as follows Section 2 delves into existing diabetes and leukemia detection methods and their limitations. The proposed LBO-ULTBASVM-based approach is explained in Section 3. Section 4 introduces the performance analysis and findings. Section 5 provides concluding remarks of the work with opportunities for future enhancement.

2. RELATED WORKS

Many recent studies have been directed in recent years for the detection of diabetes and leukemia disease using advanced ML and DL methods on biological data. Kumari et al. [9] proposed an Ensemble Soft Voting Classifier that combined random forest, Logistic Regression, and Naive Bayes for diabetes prediction using the PIMA Indian Diabetes Dataset (PIDD). The ensemble classifier achieved the best performance with 79.04% accuracy, 73.48% precision, 71.45% recall and 80.6% F1 score on the diabetes dataset. Yet, the model is not robust enough to handle and analyze additional datasets with more features. García-Ordás et al. [10] proposed a pipeline model using Variational Autoencoder (VAE) for Data Augmentation, Sparse Autoencoder (SAE) for feature augmentation, and CNN classifier to predict diabetes from PIDD. This VAE-SAE-CNN model achieved a level of accuracy of 92.31%, outperforming individual ML models. Naz et al. [11] presumed a two-phase classification module using the Synthetic Minority Oversampling Technique (SMOTE) and Sequential Minimal Optimization (SMO) for predicting diabetes. Evaluated on PIDD, this model achieved 93.07% accuracy but consumed more time. Suyanto et al. [12] developed a new framework called KMC-AE-MVMCNN for diabetes detection using K-Means Clustering (KMC), Autoencoder (AE) for dimensionality reduction, and Multi Voter Multi Commission Nearest Neighbor (MVMCNN) for classification. For binary-class PIDD, this model achieved the highest accuracy of 99.13%, while only 95.24% for the Multi-class Diabetes Type. However, this model still has a high dimensionality problem.

Olisah et al. [13] proposed a custom Twice-Growth Deep Neural Network (2GDNN) model for improving the prediction and diagnosis of diabetes mellitus. The framework applied Spearman Correlation and Polynomial Regression for the intents of feature selection and missing values handling. 2GDNN model yields 97.34% precision, 97.24% sensitivity, 97.26% F1-Score, 99.01% training accuracy, 97.25% testing accuracy on the PIDD, and 97.28% precision, 97.33% sensitivity, 97.27% F1-Score, 99.57% Train accuracy and 97.33 Test accuracy on LMCH diabetes datasets. However, this model has increased the model complexity. Annamalai et al. [14] proposed an Optimal Bi-directional LSTM (OBLSTM) model with predictive analysis and severity estimation (PASE) for diabetes detection and staging. The OBLSTM model leverages bidirectional LSTM units and is tuned via the Salp Swarm Algorithm (SSA). The two-stage OBLSTM-PASE approach achieved an accuracy of 99.53% for PIDD with a reduced false discovery rate of 0.0258 and a false positive rate of 0.0392. However, the model has a slightly increased dimensionality problem. Reza et al. [15]

proposed an improved non-linear kernel function integrating RBF and RBF city block kernels for Support Vector Machine classification of type 2 diabetes. This RBF-SVM classifier achieved precision, F1-score, accuracy and recall values of 85.5%, 83.4%, 87% and 85.2%, respectively, for PIDD. However, the model limitations exist regarding eliminating missing data in pre-processing. Al-Hameli et al. [16] proposed an Enhanced Hidden Naïve Bayes (EHNB) classifier using discretization for predicting diabetes. This model achieved 81.82% accuracy, outperforming standard HNB and other classifiers, but suffers from a slight increase in the dimensionality.

Lee et al. [17] proposed MERGE, a novel method of computational for identification of the gene expression by integrating priority information on genes' for drug sensitivity potential to drive cancer. The model is applied to gene expression and drug screening data from 30 AML patient samples across 160 chemotherapy agents and obtained a higher accuracy of 86%. However, the model has limitations in data sparsity handling. Vasighizaker et al. [18] developed a One-Class SVM (OCSVM) for predicting leukemia-causing genes. The model utilized the Gene expression data for AML and achieved 93.6% precision, 95.7% recall and 97.6% F-measure. However, the model's performance cannot be fully quantified since negative validation data is unavailable. Li et al. [19] introduced a computational framework integrating Monte Carlo Feature Selection (MCFS) and SVM to distinguish about the gene expression signatures distinguishing AML. The model selected an optimal 1159 gene feature set for evaluation and attained an accuracy of 91.6% on the AML dataset. Yet, this model has complexity limitations. Mosquera Orgueira et al. [20] proposed a random forest machine learning model called ST-123 for personalized survival prediction in AML gene expression data. Evaluated on KDM5B and LAPTM4B gene datasets, ST-123 achieved high accuracy, with c-indexes of 0.7228 and 0.6988 on the training and validation sets. However, this model has longer processing times due to class imbalances.

Karim et al. [21] proposed a novel ensemble model called LDSVM by combining Logistic regression, Decision tree, and SVM to classify leukemia cancer types accurately. Evaluated on the GSE9476 dataset containing 22285 genes, this LDSVM ensemble classifier achieved 89.8% accuracy using a hybrid approach of soft voting and 93% using hard voting, but it suffered from a high dimensionality problem. Mallick et al. [22] developed a DNN model to classify gene expression data for leukemia diagnosis. It achieved 98.21% testing accuracy, 96.59% sensitivity and 97.9% specificity, outperforming SVM, KNN and Naive Bayes classifiers. The model limitations include the high-dimensional problem leading to increased computational complexity. Angelakis et al. [23] developed a ML model called CatBoost26 to diagnose AML. The dataset that the model trained on 2177 gene expression profiles and achieved exceptionally high performance with 0.99 sensitivity, 0.99 specificity, 0.95 F1-score and 0.99 AUC. However, the model has limitations in handling the class imbalance problem. Ilyas et al. [24] developed a linear programming-based computational technique for classifying leukemia subtypes. Evaluated on the CuMiDa leukemia dataset, the model achieved 95.44% accuracy and mitigated the dimensionality curse. However, this model suffers from the poor handling of the class imbalance problem.

Analyzing the detection methods in the literature has shown that the models have been developed with higher objectives. Still, the model and computational complexities were prevalent in many models, along with the longer training and processing times, mainly due to the elevated dimensionality and imbalance in class problems. Thus, this study is performed to progress an efficient clinical decision-making approach using an advanced ML classifier to detect diabetes and leukemia accurately.

3. METHODOLOGY

The proposed LBO-ULTBASVM model aims to manage the challenges of improper feature learning, dimensionality, and imbalance in class for the detection of diabetes and leukemia accurately. Fig.1 illustrates the steps performed in the proposed approach.

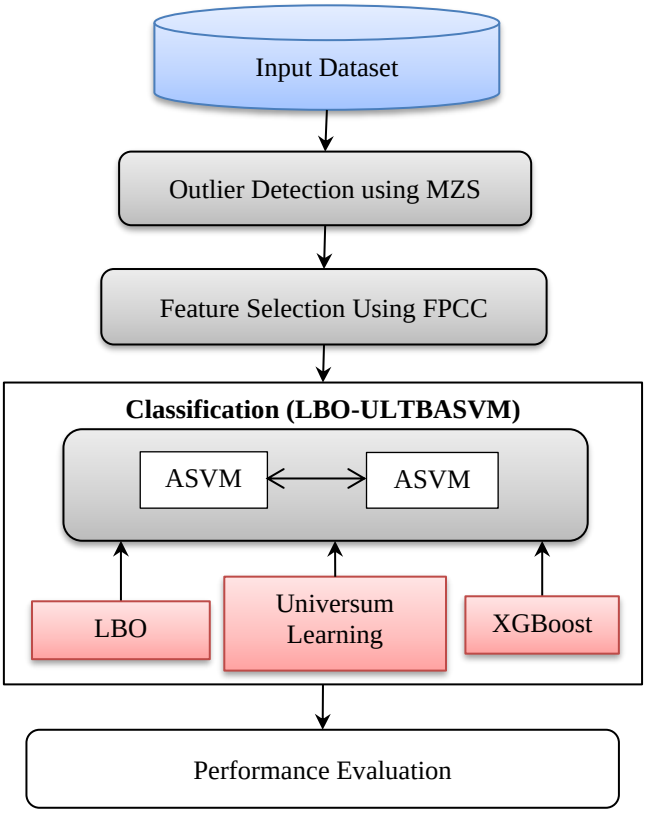

Fig.1. Proposed LBO-ULTBASVM-based model.

The proposed LBO-ULTBASVM approach integrates data pre-processing, feature selection and classification. The input biological datasets encounter issues associated with an uneven distribution of classes. A data augmentation technique is selectively utilized to mitigate the challenge of class imbalance. The initial phase of the proposed approach encompasses pre-processing the input datasets related to diabetes and leukemia disease for outlier detection and elimination. Then, the features are selected using FPCC, and the classification is performed using LBO-ULTBASVM.

3.1. Dataset Description

Biological data for disease detection includes the usage of clinical, gene expression and physiological datasets. This study utilizes two benchmark biological datasets, namely PIDD and GSE9476, to estimate the interpretation of the proposed approach.

PIMA Indian Diabetes Dataset (PIDD): This dataset, sourced from the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK), aims to predict diabetes likelihood based on diagnostic measurements. It includes 768 samples, 8 features, and a target variable. 268 samples are diabetic, while 500 samples are normal patient samples.

GSE9476: This dataset is included in the Curated Microarray Database (CuMiDa), which comprises 78 meticulously selected cancer microarray datasets. It includes 38 healthy donors and leukemic blasts from 26 AML patients with 22283 gene features. Normal hematopoietic samples included CD34+ selected cells (N = 18), unselected bone marrows (N = 10), and unselected peripheral blood (N = 10) comprise the entire dataset.

3.2. Data Pre-processing

For diabetes and leukemia datasets, pre-processing involves cleaning, transforming, and organizing the dataset to enhance its quality and compatibility with the classifiers. Tasks such as handling missing values, scaling features and addressing outliers are in the data pre-processing steps. The goal is to create a refined and standardized dataset that improves the model's accuracy and detection. The diabetes dataset is relatively small, containing many instances with limited observations in the positive class. This dataset may pose challenges for robust model training and generalization compared to the leukemia dataset. So Data Augmentation technique is applied to the diabetes datasets. The augmentation is performed by introducing random variations to the numeric features of each row. The function iterates over each column in a given row. To count the augmented variable in each row in the dataset, the augmentation factor variable is utilized. The data augmentation function is applied to each row of dataset and the process is repeated based on the specific augmentation factor. The random factor is used for order of rows and then the shuffle function is applied to the augmentation dataset. Finally the original dataset and the augmented dataset are combined on a subset of the data augmentation to make a diverse dataset for the ML models and specifically for the data where the positive class is under-represented. For more robust variations in the data model applying the factor namely augmentation factor, combining values and shuffle techniques are used for numerical features in the CSV dataset.

3.3. Outlier detection

Data cleaning in data pre-processing for diabetes and leukemia datasets involves the identification and resolve of issues like Outlier detection values. The Modified Z-Score method using Median Absolute Deviation (MAD) provides a robust measure of the spread of the data, making it less sensitive to extreme values than traditional Z-Score methods. Outliers

are identified by comparing the Modified Z-Scores (MZS) to a predetermined threshold, allowing for the effective detection and handling of outliers in a dataset.

MAD quantifies the variability within a dataset by considering the median of the absolute deviations from the dataset's median. The MAD is calculated as:

$$MAD = median(|X - median(M)|) \quad (1)$$

Here, X refers to the data point and M refers to the median of the dataset.

The MZS algorithm calculates a moving average and standard deviation from the time series data, leveraging these metrics to determine a Z-score for individual data points. The Z-score quantifies the number of standard deviations by which a data point deviates from the mean of the time series. The formula for calculating the Z-score is:

$$MZS = 0.6745 \times \left(\frac{X - median(M)}{MAD}\right) \quad (2)$$

Here, X is the data point, M is the median of the dataset.

Determining a threshold beyond which data points are identified as outliers. The threshold value used is 3.7, according to the unique characteristics of the dataset. Any data point X exhibiting a MZS surpassing the established threshold is classified as an outlier.

The MZS method using MAD provides a robust measure of the spread of the data, making it less sensitive to extreme values than traditional Z-Score methods. Outliers are identified by comparing the MZS to a predetermined threshold, allowing for the effective detection and handling of outliers in a dataset.

3.4. Feature Selection using Firefly Pearson's Correlation Coefficient

The Firefly algorithm is a meta-heuristic inspired by the flashing behaviour of real fireflies. The algorithm relies on light intensity and attractiveness, where light intensity corresponds to the brightness of a firefly, calculated using a fitness function, and determines the degree of attractiveness between fireflies.

Firefly Pearson's Correlation Coefficient method is a hybrid approach for feature selection that combines the Firefly Algorithm with Pearson's Correlation Coefficient. This technique aims to identify and rank features based on their correlation with the target variable using Pearson's Correlation Coefficient, and then optimize the selection of these features using the Firefly Algorithm.

Calculate Pearson's Correlation Coefficient for each feature in the dataset to the target variable. This measures the linear relationship between each feature and the target.

$$r_{xy} = \left(\frac{\sum_{x=1}^{n}(x - \bar{x})\,(y - \bar{y})}{\sqrt{\sum_{x=1}^{n}(x - \bar{x})^2}\ \sum_{y=0}^{n}(y - \bar{y})^2}\right) \quad (3)$$

Consider, r_{xy} as the correlation coefficient, $x\ and\ y$ are the data points, $\bar{x}\ and\ \bar{y}$ are the mean values and n is the number of data points.

In the Firefly Algorithm, the initialization of the firefly population involves randomly placing fireflies within the search space, ensuring that their positions fall within the specified bounds. The initial population of the firefly are generated and the features are assigned to the firefly. The parameter for the light intensity (attractiveness) is initialized in the initial path. The population of the firefly is initialized as $\{f_1, f_2, \cdots, f_N\}$. The initialization step of the Firefly Algorithm is to randomly place fireflies within the search space while ensuring that their positions fall within the specified bounds defined by $UB_j and\ LB_j$

$$f = LB_j + rand \times \left(UB_j - LB_j\right) \quad (4)$$

where, f is the position of firefly, $UB_j and\ LB_j$ is the upper and lower bounds solution in j^{th} dimension and $rand$ is a distributed random variable between the range [0, 1].

After this initialization, the fireflies undergo the attraction and movement phases to search for the optimal solution within the specified search space. The Firefly algorithm depends on key factors such as light intensity and the mutual attractiveness among fireflies. The movement of firefly towards another firefly is determined by the light intensity and the distance between fireflies.

Each source's light intensity is determined by the brightness of the respective firefly, computed through a specific fitness function. The brightness, linked to light intensity, governs the level of attractiveness or light intensity. The attractiveness or light intensity of each firefly is assessed using the following equation:

$$\beta(d) = \beta_0 e^{-\gamma d^2} \quad (5)$$

Here, β is denoted for the light intensity, β_0 is the attractiveness constant when the distance between the two fireflies are zero (i.e.$d_{u\,v} = 0$), γ is light intensity Co-efficient, d is the distance between the fireflies at different position. To calculate the distance between the fireflies, Euclidean distance is used:

$$ed_{u\,v} = \sqrt{\sum_{g=1}^{D}\left(f_{u,g} - f_{v,g}\right)^2} \quad (6)$$

Here, $ed_{u\,v}$ refers to the Euclidean distance between firefly u and firefly v, D refers to the total number of dimensions in the problem, representing the dimensionality of the space, g refers to the component value of position in a multi-dimensional space of the firefly along the D, $f_{u,g}$ is the position of firefly u in the g^{th} component and $f_{v,g}$ is the position of the firefly v in the g^{th} component. After computing the distance between two fireflies, if firefly u is less bright than firefly v, a light intensity takes place, resulting in the movement of firefly u toward firefly v. The movement of the firefly is represented by the following formula:

$$f_u(t+1) = f_u(t) + \beta e^{-(\gamma d_{u,v}^2)} * \left(f_v(t) - f_u(t)\right) + \alpha\epsilon \quad (7)$$

Here, $f_u(t+1)$ is the position update of firefly u at $t+1$ iterations, $f_u(t)$ is the position of firefly u at iteration t, $f_v(t)$ is the position of firefly v at iteration t, α randomized parameter with $0 \leq \alpha \leq 1$ and ϵ refers to the random vector ranges between (0, 1).

Apply the Firefly Algorithm to optimize the selection of features. The algorithm iteratively adjusts the positions of fireflies (representing features) in a search space, seeking an optimal configuration that maximizes a defined objective function.

Combining the Firefly Algorithm (FA) with Pearson's correlation coefficient for feature selection involves using the FA to search for an optimal subset of features based on their correlation with the target variable

The objective function (F) for the Firefly Pearson's Correlation Coefficient is tailored to the feature selection task and formulated as maximizing the sum or average of the ranked for the selected features.

$$F = \left(\frac{\sum_{x=1}^{n}(x-\bar{x})\,(y-\bar{y})}{\sqrt{\sum_{x=1}^{n}(x-\bar{x})^2}\ \sum_{y=0}^{n}(y-\bar{y})^2}\right) + \beta e^{-(\gamma d_{u,v}^2)} * \left(f_v(t) - f_u(t)\right) + \alpha\epsilon \quad (8)$$

The Firefly Pearson's Correlation Coefficient method combines the information from Pearson's Correlation Coefficient to rank features along with the position update and the optimization capabilities of the Firefly Algorithm to select an optimal subset of features, enhancing the efficiency of feature selection diabetes and leukemia detection tasks. Algorithm 1 illustrates the feature selection using FPCC technique

Algorithm 1: Feature Selection using FPCC

1) The population of the firefly is generated
2) The parameters are initialized
3) The position of the firefly is assigned
4) Pearson's Correlation Coefficient is calculated for each feature by using (3)
5) Evaluate fitness of each firefly based on correlation with target.
6) The light intensity and distance of each firefly are calculated.
7) Initializing the distance = 0 and calculating the light intensity for the movement of the firefly
8) While (loop < no of iterations):
    i) If $intensity\ (f_u) < \ intensity(f_v)$
    ii) Calculate the distance between the $f_u and\ f_v$.
    iii) Calculate the light intensity (attractiveness) between the fireflies.
    iv) The intensity coefficient is generated.
    v) The optimal solution using Pearson's Correlation coefficients is calculated.
    vi) Check for the position of a firefly in the search space, and update if you get a better position.

vii) Rank the fireflies according to the intensity

9) End while

10) Return the optimal solution

3.5. Ladybug Beetle Optimized Universum Learning-based Twin Boosted Adaptive Support Vector Machine (LBO-ULTBASVM) Classifier

SVM is a versatile algorithm, and the choice of the kernel and parameters depends on the characteristics of the data. SVM possess good mapping and which can also take high-dimensionality feature space to map the data. Support vector machine possesses good non-linear mapping and linear regression can be performed in the feature space, which can also take high-dimensionality feature space to map the data. The SVM is given as:

$$SVM = \min_{w,b} \frac{1}{2} \|w\|^2 \ subject\ to\ y_i\,(w.x_i + b) \geq 1\ for\ all\ i \quad (9)$$

Here, w is the weight vector, y_i is the class label, x_i is the feature vector and b is the bias parameter.

In traditional SVM, the slack variables and the margin parameter (C) is used to indirectly address misclassification by balancing the trade-off between maximizing the margin and accurately classifying the data. The SVM with the parameter is given by:

$$SVM = \min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \varepsilon_i \quad (10)$$

$$subject\ to\ y_i\,(w.x_i + b) \geq 1 - \varepsilon_{1i}\,, \varepsilon_{1i} \geq 0\ \ for\ all\ i$$

Here, C is the Parameter that controls the trade-off between maximizing the margin.

Adaptive Support Vector Machine (ASVM) is an extension of the traditional SVM that incorporates adaptability to evolving data. An Adaptive Support Vector Machine Classifier is developed to learn and adapt to the ambiguity of the data. The adaptation term involves an additional parameter δ_i, which controls the model's adaptability. The function for $ASVM$ can be written as:

$$ASVM = min_{w,b,\delta,C} \frac{1}{2} \|w\|^2 + \sum_{i=1}^{n} \delta_i \quad (11)$$

$$subject\ to\ y_i\,(w.x_i + b) \geq 1 - \delta_i,\ \ \delta_i\ \geq 0$$

Here, w is the weight, δ_i is the Adaptive variable, y_i is the class label, x_i is the feature vector.

ASVM does not use margin parameters or slack variables in its formulation. ASVM prioritizes the identification of an optimal decision boundary without explicitly addressing margins or classification errors. The adaptation variable (δ_i) introduced in enables ASVM to

adapt to evolving data patterns without directly correlating with margin or classification errors and empowers to navigate changing data distributions while maintaining its effectiveness in classification tasks.

The primary objective of TSVM is to find two optimal hyperplanes that separate the positive and negative instances while maximizing the margin and minimizing classification errors. The two parallel hyperplanes (H_1) and (H_2) are given by:

$$H_1: w_1.x + b_1 = 1 \quad (12)$$

$$H_2: w_2.x + b_2 = -1 \quad (13)$$

The marginal values for the two hyperplanes are given as:

$$C = \frac{2}{\|w_1 - w_2\|} \quad (14)$$

In TSVM, the objective function incorporates margin and slack variables to ensure that the positioning of the hyperplanes is adjusted to accurately align with the distribution of the data points. The function of twin support vector machine $(TSVM)$ can be formulated as

$$TSVM = min_{w,b_1,b_2,\varepsilon_{1i},\varepsilon_{2i}} \frac{1}{2} (\|w_1\|^2 + \|w_2\|^2) + C \sum_{i=1}^{n} (\varepsilon_{1i} + \varepsilon_{2i}) \quad (15)$$

Subject to $y_i (w_1.x_i + b_1) \geq 1 - \varepsilon_{1i},\ \varepsilon_{1i} \geq 0\ and\ y_i (w_1.x_i + b_2) \geq -1 + \varepsilon_{2i},\ \varepsilon_{2i} \geq 0$

Here, $\varepsilon_{1i}\ and\ \varepsilon_{2i}$ are the slack variables for representing the classification error for each hyperplane, $w_1\ and\ w_2$ are the weights, $b_1\ and\ b_2$ are the biases.

A kernel function is used to map the input data into a higher-dimensional feature space where a linear decision boundary can effectively separate the classes. The major Kernel functions are linear, polynomial, radial basis function (RBF) and sigmoid functions. The polynomial kernels are formulated for input (x, x_i) as follows:

Polynomial kernel: $K_P(x, x_i) = (\eta \times (x.x_i) + \sigma)^{d_p}$ (16)

Here, $K_P(x, x_i)$ is the Polynomial kernel function for $x\ and\ x_i$ inputs data points, η is a parameter representing the coefficient or weight, σ is the kernel parameter and d_p is the kernel parameters for degree of the polynomial.

By combining $TSVM$ and $ASVM$ along with the kernel parameter formulation integrates the concepts of twin hyperplanes for multi-class classification with the adaptability of ASVM to handle changing data distributions and a kernel function to map the input data into a higher-dimensional feature space. The optimization process involves finding the parameters that define the twin hyperplanes, the adaptive term, kernel parameters and the decision rule involves a voting mechanism to classify new instances. The Function of Twin Boosted Adaptive Support Vector Machine $(TBASVM)$ along with the kernel parameter is given by:

$$TASVM = min_{w_{1,2},b_{1,2},C,\varepsilon_i,\delta_i} \frac{1}{2} (\|w_1\|^2 + \|w_2\|^2) + C \sum_{i=1}^{n} (\varepsilon_{1i} + \varepsilon_{2i}) + \delta_i + K_p(\eta \times (x.x_i) + \sigma)^{d_p} \quad (17)$$

Subject to $\delta_i \geq 0 \ for\ i = 1,2,\ldots,N,\ y_i(w_1.x_i + b_1) \geq 1 - \varepsilon_{1i},\ \varepsilon_{1i} \geq 0\ and\ y_i(w_1.x_i + b_2) \geq -1 + \varepsilon_{2i},\ \varepsilon_{2i} \geq 0$

The Universum instances are denoted as:

$$U = \min_{w,b,\xi_u} \frac{1}{2}\|w\|^2 + C_u \sum_{u=1}^{m} \xi_u \quad (18)$$

Here, C_u is the parameter that represents the soft-margin for unlabelled data which minimizes the classification error for m variables and ξ_u slack variables representing the errors for unlabelled data.

The hybrid of $TASVM$ and Universum Learning forms ULTASVM, a complex approach that aims to handle uncertainty and boost classifiers. The specific implementation details, tuning mechanisms, and boosting strategy would depend on the problem's characteristics and the learning process's goals. ULTASVM aims to adaptively adjust its parameters during training, allowing it to handle better changing conditions or uncertainties in the data. The hybrid models with Universum learning are formulated as

$$ULTASVM = min_{w_{1,2},b_{1,2},C,\varepsilon_i,\delta_i,\xi_u} \frac{1}{2} (\|w_1\|^2 + \|w_2\|^2) + C \sum_{i=1}^{n} (\varepsilon_{1i} + \varepsilon_{2i}) + \delta_i + C_u \sum_{u=1}^{m} \xi_u + K_p(\eta \times (x.x_i) + \sigma)^{d_p} \quad (19)$$

Subject to $\delta_i \geq 0 \ for\ i = 1,2,\ldots,N,\ y_i(w_1.x_i + b_1) \geq 1 - \varepsilon_{1i},\ \varepsilon_{1i} \geq 0\ and\ y_i(w_1.x_i + b_2) \geq -1 + \varepsilon_{2i},\ \varepsilon_{2i} \geq 0$

ULTBASVM is designed to provide a flexible and robust solution for boosted classification tasks with an adaptive regularization mechanism for best performance using XGBoost to improve the ULTASVM model. The output from ULTASVM is provided as input to the XGBoost, which involves exploring the combinations of optimum hyperparameters that resultant to produce the best performing model. The process typically includes defining a parameter grid, performing cross-validation, and selecting the parameters that maximize the chosen evaluation metric. The objective function in XGBoost consists of a loss function that quantities the difference between predicted and true values and a regularization term that penalizes complexity. The objective function (Obj) is formulated as:

$$Obj(\theta) = \sum_{i=1}^{n} \mathbb{L}(y_l.\widehat{y_l}) + \Omega(p_t) \quad (20)$$

Here, n is the training instances number, θ is the model set parameter, y_l is the true label, $\widehat{y_l}$ is the value predicted $\mathbb{L}$ is defined as the loss function and $\Omega(p_t)$ is the complexity penalty in the regularization.

The XGBoost uses various hyperparameters for parameter tuning for the model. The hyperparameters include Learning Rate, Maximum Depth, estimators, subsample and the random state. Therefore, the proposed ULTBASVM becomes

$$\text{ULTBASVM} = min_{w_{1,2},,b_{1,2},C,\varepsilon_i,\delta_i,\xi_u} \frac{1}{2}\left(\|w_1\|^2 + \|w_2\|^2\right) + C\sum_{i=1}^{n}(\varepsilon_{1i} + \varepsilon_{2i}) + \delta_i + C_u\sum_{u=1}^{m}\xi_u + K_p(\eta \times (x.x_i) + \sigma)^{d_p} + \sum_{i=1}^{n}\mathbb{L}(y_l.\hat{y_l}) + \Omega(p_t) \quad (21)$$

LBO: It is the inspired by the coordinated movement of ladybugs in nature to find a location with the most heat. The algorithm also helps to avoid the local minimum problems and speeds up the flow of the algorithm. The algorithm is used to optimize the ULTBASVM parameters namely marginal parameters and the kernel function. The algorithm involves evaluating and sorting the population. The population undergoes position updates and is re-evaluated. This cycle of updating the population and evaluating is repeated until the optimal solution is identified. Let p be the position of the ladybug and the population parameter as $\{l_1, l_2, \cdots, l_N\}$. The population is composed of l_{max} ladybugs, with the condition $l(0) \geq l_{max}$ and the optimal objective function is determined. The position of ladybug is obtained by:

$$l_p = l_r(h) + rand \times \left(l_s(h) - l_r(h)\right) + rand \times \left(l_s(h) - l_{s-1}(h)\right) \times |TC|^{-\frac{h}{l(h)}} \times l_s(h) \quad (22)$$

Here, l_p refers to the position of the ladybug, $l_r(h)$ represents the current position of the r^{th} ladybug in the h iteration, $l_s(h)$ represents the position of s^{th} ladybug in the h iteration, $l_{s-1}(h)$ refers to the position of the second neighbour ladybug in the h iteration and $rand$ is a random number uniformly distributed between 0 and 1, TC represents the ratio of the total cost of the ladybird.

The TC function measures the attractiveness of a specific location in the search space for a ladybug based on the objective function value at that location. It calculates the ratio of the objective function value at the current position ($l_r(h)$) of a ladybug to the sum of objective function values at the positions of all ladybugs within its neighbourhood. The Total cost (TC) function is given as

$$TC = \frac{fit(l_r(h))}{\sum_{N=1}^{N^h} fit(l(h))} \quad (23)$$

Here, $fit(l_r(h))$ refers to the objective function value at the current position of the r^{th} ladybug, N^h refers to the total number of ladybug at h iteration and $\sum_{N=1}^{N^h} fit(l(h))$ refers to the Sum of objective function values at the positions of all ladybugs within the neighbourhood of the ladybug l_p.

The TC function evaluates the attractiveness of a ladybug's current position relative to its neighbourhood based on the objective function values, guiding the ladybugs towards promising areas of the search space during the optimization process.

Instead of using the TC function from LBO, the ULTBASVM objective function is employed to evaluate the attractiveness of ladybugs' positions for optimization. Ladybugs move towards positions that possess maximum accuracy for the ULTBASVM objective function, effectively optimizing the margin parameters (C, C_u) and the polynomial kernel function. The position of ladybug along with ULTBASVM is obtained by:

$$l_p = l_r(h) + rand \times \left(l_s(h) - l_r(h) \right) + rand \times \left(l_s(h) - l_{s-1}(h) \right) \times |\text{ULTBASVM}|^{-\frac{h}{l(h)}} \times l_s(h) \quad (24)$$

The equation is derived from the replacing TC in (22) by ULTBASVM from (21). By exchanging the total cost function with the ULTBASVM objective function, the LBO algorithm guides the search towards regions of the parameter space that lead to improved classification performance and model complexity control.

The ULTBASVM objective function captures the performance and complexity aspects of the ULTBASVM classifier, incorporating margin parameters (C, C_u) and the polynomial kernel (K_p) function. By employing the ULTBASVM objective function, ladybugs are guided towards positions in the search space that lead to maximum accuracy and optimal model complexity, as determined by the ULTBASVM classifier.

In the process of searching for a warm place, it's common for ladybugs to become disoriented and vanish. They might stray from the group and perish due to the cold. In the LBO algorithm, this phenomenon of updating population size of ladybugs disappearing during the search is mathematically modelled and the number of ladybugs present at each step of the search is determined as follows:

$$l_r(h+1) = \left(l_r(h) - rand \times l(h) \left(\frac{NFE}{NFE_{max}} \right) \right) \quad (25)$$

Here, $l_r(h+1)$ represents the new position of the r^{th} ladybug in the $(h+1)$ iteration, NFE denotes the function evaluations number, and NFE_{max} represents the maximum allowable number of function evaluations.

Thus, (22) is used if the termination condition for LBO algorithm is based on the number of function evaluations

If the number of iterations serves as the termination condition for the algorithm, the algorithm proceeds to calculate the new number of ladybugs in each iteration using the following method:

$$l_r(h+1) = \left(l_r(h) - rand \times l(h) \left(\frac{h}{h_{max}} \right) \right) \quad (26)$$

Here, $l_r(h+1)$represents the new position of the r^{th} ladybug in the $(h+1)$ iteration, $l_r(h)$ represents the current position of the r^{th} ladybug in the h iteration, h denotes the current iteration number and h_{max} represents the maximum number of iterations. Algorithm 2 summarizes the LBO for tuning ULTBASVM.

Algorithm 2: LBO for tuning ULTBASVM

Start

1) The parameters are initialized for the ULTBASVM: Margin parameters $(C, C_u,)$Kernel parameters η, σ and d_p
2) Initialize the ladybug population positions
3) Set the maximum margin threshold l_{max}
4) For r^{th} position for the number of the ladybug:
    (a) Calculate the ULTBASVM objective function with the sum of ladybugs
    (b) A random number is created
    (c) If $rand > l_r$,
        (i) Update the ladybug position, in comparison to the other ladybug
        (ii) Calculate the ULTBASVM objective function sum for the updated position
        (iii)Update the objective function sum for the updated position.
        (iv)Update the position of the r^{th} ladybug $l_r(h+1)$
        (v) Calculate the function for the new population
    (d) End if
    (e) If $l_r(0) \geq l_{max}\ do$
        1. $l_r(h+1) = l_{max}$
    (f) End if
5) End for

Stop

Therefore, the cost function for the proposed model is calculated based on the margin, kernel and kernel parameters of the ULTBASVM classifier. Algorithm 3 summarizes the proposed classifier.

Algorithm 3: ULTBASVM classifier

Start
Initialize Parameters:
  Set ULTBASVM and XGBoost parameters
  Initialize population of ladybugs with random positions
  Set termination conditions- h_{max}, NFE_{max}
ULTBASVM Training with XGBoost:
Train ULTBASVM model using XGBoost.

Explore combinations of optimum hyperparameters using XGBoost for parameter tuning.
Formulate the function in XGBoost, consisting of a loss function and a regularization term
Evaluate model performance
LBO Optimization:
Evaluate and sort population of ladybugs based on ULTBASVM model performance.
Repeat until termination condition is met:
Update the position of each ladybug using the movement strategy:
if termination condition based on NFE:
Calculate new position using (25)
else if termination condition based on iterations:
Calculate new position using (26)
Final Model Configuration:
  Determine optimal parameters obtained from LBO optimization
  Configure ULTBASVM classifier with optimized parameters
  Return final configuration of ULTBASVM classifier
Stop

The algorithm aims to find the optimal solution for the best parameter tuning. The parameters are adjusted for the exploitation and exploration characteristics. Ladybugs iteratively adjust their positions based on the attractiveness of their current positions relative to their neighbourhoods, as assessed by the ULTBASVM objective function. Positions that result in improved classification performance and well-controlled model complexity are favoured, and ladybugs move towards such positions during the optimization process. This approach allows for more effective exploration of the parameter space and facilitates the discovery of solutions that yield superior performance and accuracy in classification tasks. This returns the final configuration of the ULTBASVM classifier for implementing the clinical decision-making model for diabetes and Leukemia classification.

4. RESULT AND DISCUSSION

The suggested LBO-ULTBASVM-based model for the disease detection problem is assessed using benchmark biological datasets. The implementations use Python programming on a PyCharm tool on a system with an Intel Core i5 processor, Windows 10 OS with 8GB RAM and 512GB SSD. Accuracy, Precision, Recall, F-Measure, and Processing Time are used for the evaluation. The attained results of the interpreted framework are equated with the different existing approaches. Table 1 presents the results for the proposed approaches obtained by performing step-by-step derivations for both the PIDD and GSE9476 biological datasets.

Table.1. Accuracy of Proposed Approach in Each Step of Modification

Algorithm	PIDD	GSE9476
SVM	89.45	84.23
TSVM	90.99	86.51
ASVM	91.33	88.02
TASVM	92.90	88.78
ULTASM	93.71	90.88
ULTBASVM	94.95	92.30
LBO-ULTBASVM	**96.68**	**94.36**

The above table shows the accuracy of the proposed LBO-ULTBASVM is higher when compared to prior classification models. The model performance has been increased by 1%-2% and has shown the best results in both datasets. Therefore, the performance of the proposed model is justified.

To evaluate the suggested methodology further, they are equated with the methods used in the previous studies. Since the previous methods have used different datasets for diabetes and leukemia in different experimental conditions, directly equating the results will not be ideal. Hence, for an equal comparison, the methods depicted in these studies are executed in the same environment as the presumed approach over the PIDD and GSE9476 datasets. All the methods in the related works section are utilized for both diabetes and leukemia disease datasets to evaluate their diversity. Table 2 lists the results of the suggested approach against the literature review obtained for PIDD.

Table.2. Performance of LBO-ULTBASVM vs. literature methods for diabetes-PIDD

Methods/ Metrics	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Processing time (s)
Ensemble Soft Voting [9]	PIDD	79.04	73.48	71.45	80.6	144.34
VAE-SAE-CNN [10]	PIDD	92.31	89.45	87.34	88.38	140.11
SMOTE–SMO [11]	PIDD	96.07	90.22	89.98	90.1	150.32
KMC-AE-MVMCNN [12]	PIDD	94.13	91.29	91.90	91.59	144.32
2GDNN [13]	PIDD	95.25	97.34	90.23	97.26	145.79
OBLSTM-PASE [14]	PIDD	95.53	83.12	86.81	84.92	143.72
RBF-SVM [15]	PIDD	85.5	83.4	87	85.2	145.63
EHNB [16]	PIDD	81.82	93.67	91.98	92.82	139.82
LBO -ULTBASVM	**PIDD**	**96.68**	**98.28**	**97.79**	**98.03**	**138.16**

From Table 2, the LBO-ULTBASVM model outperforms existing models and achieves higher accuracy, recall, and f1-measure, along with efficient processing time. An evaluation of individual, hybrid, and fusion models in the literature indicates that the FPCC and LBO–ULTBASVM model surpasses other models. The proposed model LBO – ULTBASVM increased accuracy by 17.64%, 4.37%, 0.61%, 2.55%, 1.43%, 1.15%, 11.18% and 14.86% for Ensemble Soft Voting, VAE-SAE-CNN, SMOTE-SMO, KMC-AE-MVMCNN, 2GDNN, OBLSTM-PASE, RBF-SVM and EHNB respectively. The Processing time is reduced by 4.48%, 1.41%, 8.78%, 4.47%, 5.54%, 4.02%, 5.40% and 1.20% for Ensemble Soft Voting, VAE-SAE-CNN, SMOTE-SMO, KMC-AE-MVMCNN, 2GDNN, OBLSTM-PASE, RBF-SVM and EHNB respectively. Overall, LBO - ULTBASVM shows an increase of approximately 9.32% in accuracy and decrease of approximately 4.36% in processing time. A detailed analysis of various Classification models, both individual and hybrid, on diabetes datasets confirms the significantly superior performance and advantages of the proposed FPCC and LBO–ULTBASVM model, boasting accuracy and a reduced processing time.

Similarly, Table 3 shows the results of the suggested technique against the literature review obtained on Gene expression datasets for Leukemia disease detection.

Table.3. Performance of LBO-ULTBASVM vs. literature methods for Leukemia datasets

Methods/ Metrics	**Dataset**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F-measure (%)**	**Processing time (s)**
MERGE [17]	GSE108004	90.76	93.6	95.7	97.6	378.23
OCSVM [18]	GSE9476	85.18	99.06	95.7	97.6	355.12
ST-123 [19]	GSE45249	94.06	92.00	90.46	91.22	343.67
MCFS- SVM [20]	GSE37642	72.28	80.83	83.55	82.17	392.84
LDSVM [21]	GSE9476	94.08	90.88	88.01	89.42	390.04
DNN [22]	GSE32474	93.21	89.92	89.14	89.53	377.45
CatBoost26 [23]	GSE26312	93.76	90.34	90.05	99.69	402.34
Linear Programming [24]	GSE9476	93.44	91.11	89.99	90.55	433.5
LBO - ULTBASVM	**GSE9476**	**94.36**	**99.73**	**99.70**	**99.71**	**333.06**

The results in Table 3 show that the LBO–ULTBASVM model outperforms existing models and achieves higher accuracy, recall, and f1-measure, along with efficient processing time. The proposed FPCC and LBO–ULTBASVM model increased accuracy by 3.6%, 9.18%, 0.3%, 22.08%, 0.28%, 1.15%, 0.6% and 0.92% for MERGE, OCSVM, ST-123, MCFS-SVM, LDSVM, DNN, CatBoost26 and Linear Programming respectively and reduced processing time by 11.92%, 6.21%, 3.09%, 15.19%, 14.63%, 11.78%, 17.23% and 23.17% for MERGE, OCSVM, ST-123, MCFS-SVM, LDSVM, DNN, CatBoost26 and Linear Programming respectively. Utilizing Universum learning with performance-boosting and the FPCC has greatly reduced the class imbalance and high dimensionality issues and contributed to this performance improvement. Overall, detailed analysis of various hybrid Classification

models, on Leukemia datasets confirms the significantly superior performance and advantages of the proposed FPCC and LBO–ULTBASVM model, boasting accuracy and a reduced processing time.

5. CONCLUSION

This paper presented a hybrid LBO-ULTBASVM-based approach for diabetes and leukemia disease detection through biological data analysis. The proposed approach includes MZS for outlier detection, FPCC for feature/gene selection and LBO-ULTBASVM model for classification processes. The LBO-ULTBASVM classification model presents a promising approach by combining the power of the LBO metaheuristic, Universum Learning, and XGBoost to improve the Twin Adaptive SVM classifier. This integrated model reduces the high dimensionality and class imbalance problems and accurately detects diabetes and leukemia. LBO-ULTBASVM model achieves higher accuracy, precision, recall, F1-measure, and less processing time. It achieved an accuracy of 96.68% for diabetes detection with 138.16 seconds processing time and 94.36% accuracy for leukemia detection with 333.06 seconds processing time. Thus, the LBO-ULTBASVM classification model provides promising results for detecting diabetes and leukemia by solving the high dimensionality and class imbalance problems. The proposed LBO-ULTBASVM has identified the AML accurately; therefore, the adaptability of the proposed approach for classifying other types of leukemia from diverse biological datasets will be examined in the future.

REFERENCES

1. González-Lao, E., Corte, Z., Simón, M., Ricós, C., Coskun, A. B. D. U. R. R. A. H. M. A. N., Braga, F., ... & Sandberg, S. (2019). Systematic review of the biological variation data for diabetes related analytes. *Clinica Chimica Acta*, *488*, 61-67.
2. U. Galicia-Garcia, A. Benito-Vicente, S. Jebari, A. Larrea-Sebal, H. Siddiqi, K. B. Uribe, and C. Martín, "Pathophysiology of type 2 diabetes mellitus," *International journal of molecular sciences*, vol. *21*(17), pp. 6275, 2020
3. J. A. B. Bispo, P. S. Pinheiro, and E. K. Kobetz, "Epidemiology and etiology of Leukemia and lymphoma," *Cold Spring Harbor perspectives in medicine*, vol. *10*(6), 2020.
4. Y. Dong, O. Shi, Q. Zeng, X. Lu, W. Wang, Y. Li, and Q. Wang, "Leukemia incidence trends at the global, regional, and national level between 1990 and 2017," *Experimental hematology & oncology*, vol. *9*, pp. 1-11, 2020.
5. H. E. Williams, C. R. Howell, W. Chemaitilly, C. L. Wilson, S. E. Karol, V. G. Nolan, and K. K. Ness, "Diabetes mellitus among adult survivors of childhood acute lymphoblastic leukemia: a report from the St. Jude Lifetime Cohort Study," *Cancer*, vol. *126*(4), pp. 870-878, 2020.
6. J. J. Khanam, and S. Y. Foo, "A comparison of machine learning algorithms for diabetes prediction," *Ict Express*, vol. *7*(4), pp. 432-439, 2021.
7. M. A. Alsalem, A. A. Zaidan, B. B. Zaidan, M. Hashim, H. T. Madhloom, N. D. Azeez, and S. Alsyisuf, "A review of the automated detection and classification of acute Leukemia: Coherent taxonomy, datasets, validation and performance

measurements, motivation, open challenges and recommendations," *Computer methods and programs in biomedicine*, vol. pp. *158*, 93-112, 2018.

8. S. Safiri, and A. Nikoofard, "Ladybug Beetle Optimization algorithm: application for real-world problems," *The Journal of Supercomputing*, vol. *79*(3), pp. 3511-3560, 2023.
9. S. Kumari, D. Kumar, and M. Mittal, "An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier," *International Journal of Cognitive Computing in Engineering*, vol. *2*, pp. 40-46, 2021
10. M. T. García-Ordás, C. Benavides, J. A. Benítez-Andrades, H. Alaiz-Moretón, and I. García-Rodríguez, "Diabetes detection using deep learning techniques with oversampling and feature augmentation," *Computer Methods and Programs in Biomedicine*, vol. *202*, pp. 105968, 2021.
11. H. Naz, and S. Ahuja, "SMOTE-SMO-based expert system for type II diabetes detection using PIMA dataset," *International Journal of Diabetes in Developing Countries*, vol. *42*(2), pp. 245-253, 2022.
12. S. Suyanto, S. Meliana, T. Wahyuningrum, and S. Khomsah, "A new nearest neighbor-based framework for diabetes detection," *Expert Systems with Applications*, vol. *199*, pp. 116857, 2022
13. C. C. Olisah, L. Smith, and M. Smith, " Diabetes mellitus prediction and diagnosis from a data pre-processing and machine learning perspective," *Computer Methods and Programs in Biomedicine*, vol. *220*, pp. 106773, 2022
14. R. Annamalai, and R. Nedunchelian, "Design of optimal bidirectional long short term memory based predictive analysis and severity estimation model for diabetes mellitus," *International Journal of Information Technology*, vol. *15*(1), pp. 447-455, 2023.
15. M. S. Reza, U. Hafsha, R. Amin, R. Yasmin, and S. Ruhi, "Improving SVM performance for type II diabetes prediction with an improved non-linear kernel: Insights from the PIMA dataset," *Computer Methods and Programs in Biomedicine Update*, *4*, pp. 100118, 2023.
16. B. A. Al-Hameli, A. A. Alsewari, S. S. Basurra, J. Bhogal, and M. A Ali, (2023). "Diabetes disease prediction system using HNB classifier based on discretization method," *Journal of Integrative Bioinformatics*, vol. *20*(1), pp. 20210037, 2023.
17. S. I. Lee, S. Celik, B. A. Logsdon, S. M. Lundberg, T. J. Martins, V. G. Oehler, and P. S. Becker, "A machine learning approach to integrate big data for precision medicine in acute myeloid leukemia," *Nature communications*, vol. *9*(1), pp. 42, 2018.
18. A. Vasighizaker, A. Sharma, and A. Dehzangi, "A novel one-class classification approach to accurately predict disease-gene association in acute myeloid leukemia cancer," *PloS one*, vol. *14*(12), pp. e0226115, 2019.
19. J. Li, L. Lu, Y. H. Zhang, Y. Xu, M. Liu, K. Feng, and Y. D. Cai, "Identification of Leukemia stem cell expression signatures through Monte Carlo feature selection strategy and support vector machine," *Cancer Gene Therapy*, vol. *27*(1-2), pp. 56-69, 2020.

20. A. Mosquera Orgueira, A. Peleteiro Raíndo, M. Cid López, J. A. Díaz Arias, M. S. González Pérez, B. Antelo Rodríguez, and J. Luis Bello López, "Personalized survival prediction of patients with acute myeloblastic leukemia using gene expression profiling," *Frontiers in Oncology*, vol. *11*, pp. 657191, 2021.
21. A. Karim, A. Azhari, M. Shahroz, S. B. Belhaouri, and K. Mustofa, "LDSVM: Leukemia cancer classification using machine learning," *Comput. Mater. Contin*, vol. *71*(2), pp. 3887-3903, 2021.
22. P. K. Mallick, S. K. Mohapatra, G. S. Chae, and M. N. Mohanty, "Convergent learning–based model for leukemia classification from gene expression," *Personal and Ubiquitous Computing*, vol. *27*(3), pp. 1103-1110, 2023.
23. A. Angelakis, I. Soulioti, and M. Filippakis, "Diagnosis of acute myeloid leukemia on microarray gene expression data using categorical gradient boosted trees," *Heliyon*, vol. *9*(10), 2023.
24. M. Ilyas, K. M. Aamir, S. Manzoor, and M. Deriche, "Linear programming-based computational technique for leukemia classification using gene expression profile," *Plos one*, vol. *18*(10), pp. e0292172, 2023.