Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 8(10): 24274–24309.
DOI: 10.3934/math.20231238
Received: 27 April 2023
Revised: 08 July 2023
Accepted: 01 August 2023
Published: 14 August 2023

Research article

Efficient thyroid disorder identification with weighted voting ensemble of super learners by using adaptive synthetic sampling technique

Noor Afshan[1], Zohaib Mushtaq[2], Faten S. Alamri[3,*], Muhammad Farrukh Qureshi[4], Nabeel Ahmed Khan[4] and Imran Siddique[5]

[1] Department of Software Engineering, Faculty of Computer Science, Lahore Garrison University, Lahore 54000, Pakistan

[2] Department of Electrical Engineering, CET, University of Sargodha, Sargodha 40100, Pakistan

[3] Department of Mathematical Sciences, College of Science, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

[4] Department of Electrical Engineering, Riphah International University, Islamabad 44000, Pakistan

[5] Department of Mathematics, University of Management and Technology, Lahore 54770, Pakistan

* **Correspondence:** Email: fsalamri@pnu.edu.sa.

Abstract: There are millions of people suffering from thyroid disease all over the world. For thyroid cancer to be effectively treated and managed, a correct diagnosis is necessary. In this article, we suggest an innovative approach for diagnosing thyroid disease that combines an adaptive synthetic sampling method with weighted average voting (WAV) ensemble of two distinct super learners (SLs). Resampling techniques are used in the suggested methodology to correct the class imbalance in the datasets and a group of two SLs made up of various base estimators and meta-estimators is used to increase the accuracy of thyroid cancer identification. To assess the effectiveness of our suggested methodology, we used two publicly accessible datasets: the KEEL thyroid illness (Dataset1) and the hypothyroid dataset (Dataset2) from the UCI repository. The findings of using the adaptive synthetic (ADASYN) sampling technique in both datasets revealed considerable gains in accuracy, precision, recall and F1-score. The WAV ensemble of the two distinct SLs that were deployed exhibited improved performance when compared to prior existing studies on identical datasets and produced higher prediction accuracy than any individual model alone. The suggested methodology has the potential to increase the accuracy of thyroid cancer categorization and could assist with patient diagnosis and treatment. The WAV ensemble strategy computational complexity and the ideal choice of base estimators in SLs continue to be constraints of this study that call for further investigation.

Keywords: thyroid cancer; cancer detection; hypothyroid; machine learning; super learners
Mathematics Subject Classification: 62J02, 62J99

1. Introduction

Thyroid cancer is one of the most common endocrine malignancies, accounting for approximately 3.4% of all new cancer cases globally [1,2]. It is estimated that there were 567,233 new cases of thyroid cancer and 41,071 deaths from the disease in 2020 alone [3]. Thyroid cancer is particularly prevalent in women, with a female-to-male incidence ratio of 3:1 [4]. Figure 1 shows the hypothyroid infection in various countries as percentage of the population. Risk factors for developing thyroid cancer include exposure to ionizing radiation, family history of thyroid cancer and certain genetic mutations [5]. This growing incidence has been contributed to numerous aspects, such as increased exposure to ionizing radiation, environmental pollutant and improved diagnostic techniques such as high-resolution ultrasound and fine-needle aspiration biopsy [6, 7]. Thyroid gland has two hormones, thyroxine (T4) and triiodothyronine (T3) and dysregulation of thyroid hormones can result in several pathological conditions, including hypothyroidism, hyperthyroidism and thyroid cancer [8]. Noninvasive technique such as ultrasound computed tomography (CT) scans and an invasive technique such as fine-needle aspiration biopsy (FNAB) are used for detection of thyroid cancer [9–11]. Ultrasound can distinguish between solid and cystic nodules and can also identify cancers, such as irregular borders, microcalcifications and increased vascularity of thyroid cells. If an ultrasound reveals a suspicious nodule, an FNAB may be performed to obtain a tissue sample for microscopic examination. Early diagnosis is very important for improving the prognosis and reducing the mortality rates associated with this malignancy. Recent advances in machine learning (ML) techniques have the potential to significantly improve the accuracy and efficiency of thyroid cancer classification, aiding clinicians in making better-informed treatment decisions.

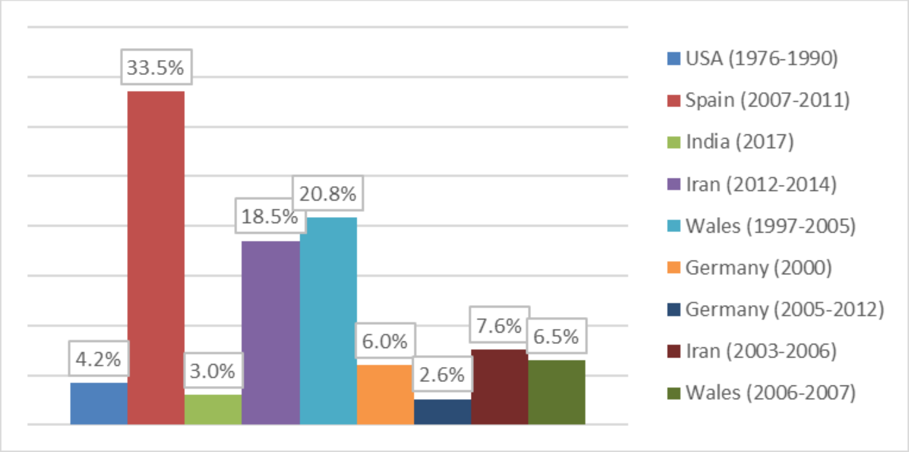


Figure 1. Presence of hypothyroid infection in various countries in the percentage.

Machine learning techniques have demonstrated their utility in various aspects of cancer research and clinical practice, such as disease diagnosis, prognosis and treatment selection [12]. In the context of thyroid cancer, ML algorithms have been employed to analyze a variety of data types, including imaging data, genomic data and clinical data, to provide valuable insights into the classification and prediction of the disease [13–17]. For instance, ML techniques have shown promising results in the classification of thyroid nodules using ultrasound images [18, 19], prediction of aggressive tumor features based on clinical and histopathological data [20, 21] and molecular classification of thyroid cancer subtypes using genomic data [22, 23]. The integration of ML techniques into thyroid cancer diagnostics and treatment decision-making has the potential to enhance patient care by improving the accuracy of diagnoses, reducing unnecessary interventions and facilitating personalized treatment planning. However, despite these promising advancements, challenges remain in terms of data quality, model interpretability and clinical implementation, warranting further research and development in this area.

After the pre-processing part, handling the class imbalance issue effectively for both datasets is a top concern. In machine learning, imbalanced datasets occur frequently when there are significantly less examples in one class than in another. This can have a negative impact on the accuracy of machine learning models, which is especially problematic for marginalized groups. The two datasets have been used in this research work. The first dataset comprises the three classes in the target variable, with the representation of normal, hypo and hyperthyroidism. The total instances for each target class are, for normal total of 166, hypothyroidism consists of 6666 samples and hypothyroidism includes only 368 samples. Similarly, the second dataset includes two classes, 3,481 samples labelled as P and 291 as N updated as suggested. It is clearly shown that both datasets include imbalance classes that can directly affect the performance and accuracy of the proposed model, due to the limited number of training samples for the specific target variable. Therefore, in this study, we implemented adaptive synthetic (ADASYN) sampling to resample the minority class target variables. Table 2 includes a detailed discussion about the total number of samples for the original and ADASYN-generated datasets.

The last part of this research study focused on the implementation of the ensembling technique for the two implemented super learners. Although super learning itself is an ensembling technique where multiple base meta estimators have been used to combine the predictions of the models. Super learners are a sort of ensemble learning in which the predictions of numerous models are combined to improve overall performance. Cross-validation is used by the super learner method to estimate the performance of many machine learning models. By lowering bias and variance and eliminating parametric assumptions, the super learner method can increase the accuracy of machine learning models. It can also assist to avoid overfitting and increase model generalization. In this study, two super learners were implemented that again undergoes the voting ensemble, to consecutively improve the accuracy and performance of the proposed approach on both datasets.

The main contributions of this work are below:

- Novel ensemble modeling approach utilizing two super learners, each containing three distinct classifiers, to boost classification performance and reduce model variance.
- Various preprocessing and feature selection techniques employed, including feature importance techniques, dimensionality reduction methods, class imbalance handling, outlier detection and feature standardization, to streamline the datasets and identify the most relevant features for thyroid disease classification.
- Class imbalance issues were addressed using the adaptive synthetic (ADASYN) sampling technique, oversampling the minority class to ensure equal representation of all classes.

The paper is structured into several sections, starting with a review of the relevant literature and prior research on thyroid disease classification in Section 2. Section 3 describes the methodology employed in this study, including data acquisition, preprocessing, feature importance, outlier detection, class imbalance handling and ensemble modeling. Section 4 presents the findings of the study, including a comparison with existing works. Section 5 provides an interpretation and analysis of the results. Finally, Section 6 summarizes the study's main contributions, limitations and potential for future research.

2. Related works

Several studies have used ML techniques such as support vector machines (SVM), artificial neural networks (ANN) and deep learning algorithms such as convolutional neural networks (CNNs) for detection and classification of thyroid cancer. One area where ML has shown promise is in the diagnosis of thyroid nodules, which is crucial for accurate and timely treatment planning. The study [24] proposed a deep learning technique that is based on a deep convolutional neural network (CNN) to distinguish between benign and malignant thyroid nodules using ultrasound images. The dataset consisted of 1,000 ultrasound images of thyroid nodules, which were divided into training, validation and testing sets. The CNN model achieved an accuracy of 87.6% on the testing set and demonstrated high sensitivity in detecting malignant nodules. Another study [25], employed a machine learning approach to predict the presence of the BRAF mutation in cancerous thyroid nodules. The researchers used 96 ultrasonic images of thyroid nodules and extracted 86 radiomic features. They utilized three different models, namely linear regression (LR), support vector machine (SVM) and random forrest (RF), to predict the likelihood of the BRAF mutation being present. Another study [26] proposed a thyroid nodule classification system based on feature fusion and deep learning techniques. The dataset consisted of 5,310 ultrasound images of thyroid nodules and the proposed system achieved high accuracy (95.2%), sensitivity (93.1%) and specificity (96.8%) using a combination of CNN and LSTM networks. In the research conducted in [27], Chen et al. utilized the LASSO technique along with a LR model to pick out the ultrasonic characteristics associated with malignant thyroid nodules. Subsequently, they employed RF to categorize the malignant thyroid nodules. By using LLR in conjunction with RF, they achieved the highest level of accuracy, which was 82%.

ML has also been applied to predict the risk of malignancy in thyroid nodules using radiomics features extracted from CT images. ML has also been applied to prognostic modeling in thyroid cancer, which is essential for personalized treatment planning and improved patient outcomes. The study described in [28] used two machine learning techniques, namely SVM and RF, to detect thyroid disorders using the thyroid dataset provided. The SVM model achieved 91% accuracy, while the RF model achieved 89%. In the study done in [29], the research aimed to forecast thyroid disease, categorizing it into two types: hypothyroid and euthyroid. The assessment criteria adopted in the research encompassed accuracy, precision, recall, F1-score, ROC-AUC, confusion matrix and classification. The random forest classifier stood out as the most effective approach, achieving a success rate of 99.5%. The study emphasized the capacity of machine learning algorithms in detecting and diagnosing thyroid disease in its initial stages. In another work [30], The model employed in the study was an in symbol of homogenous ensembles that combined multiple attributes selection approaches. The findings of the study demonstrated that the proposed method achieved impressive accuracy of 99.6% with surpassed the other state of the art approaches. The algorithm emerged as the best technique used in the study. Another study [31] proposed an artificial neural network (ANN) model to differentiate between benign and malignant nodules and improve the accuracy of objective diagnosis based on ultrasound (US) images. The ANN accurately predicted 82.3% of thyroid cancer cases with an AUC value of 0.818 and an accuracy rate of 84.5%.

In another study [32], it was observed that SVM was more effective than RF in identifying thyroid conditions. The study employed ML classifiers to predict the presence of thyroid disorders. To enable algorithms to identify the likelihood of patients developing a particular disease, data preparation techniques were implemented to simplify the data. Disease prediction using machine learning is a common practice and several methods are employed by scientists, such as SVM, DT, LR, ANN and KNN, to predict the likelihood of a patient acquiring thyroid disease. In this study [33], clinical datasets were employed to evaluate and compare the performance of three classifiers: SVM, NB and DT. SVM is widely utilized in machine learning. The study [34] categorized thyroid disease into three groups based on data i.e. overactive thyroid and hypothyroidism. The study implemented several classification methods, including SVM, DT, RF, NB, LR, KNN, LDA and MLP. The most accurate classifiers was RF, achieving 89% accuracy.

In another study [35], researchers employed three ML techniques ANN, RF and SVM to identify thyroid texture. The researchers created 30 attributes based on spectral energy using autoregressive modeling for a 2D thyroid ultrasound image variation to train the classifiers. The characteristics of thyroid tissues were illustrated using image-based features instead of text-based descriptors. When the three techniques were combined, the accuracy rate was around 90%. In [36], the authors used data mining techniques using python to create algorithms for identifying thyroid illness types. It has enabled cost effective thyroid diagnostic reports to be available to patients. Two well-known systematic attribute selection techniques, namely sequential forward and sequential backward selection, were utilized. The evolutionary method was used as a popular strategy for picking features in nonlinear optimization problems. The SVM was employed to detect hypothyroidism.

In a cross-sectional study [37], a classification algorithm was developed by integrating SVM, MLP, CHAID and iterative dichotomiser-3. To address dataset imbalance issues, classification methods, bootstrap aggregating (Bagging) and boosting procedures were utilized, which improved the classification outcomes. The study revealed that SVM bagging produced 100% precision and specificity, 73.33% recall and 84.62% F-measure. In a different study [38], the attribute partitioning criteria for detecting thyroid disease were determined using DT. The authors aimed for an accuracy rate of 99.89% and compared the diagnostic results using DT, SVM and NB methodologies. In another study [39], DT, KNN and SVM were used to evaluate the risk of thyroid illness based on a patient's medical history using various ML methods for disease-prevention diagnostics.

Table 1 compares existing studies on thyroid disease detection using various datasets for evaluation. For our study, we chose a well-known UCI dataset. While previous studies achieved high accuracy in detecting and classifying thyroid disease, there has been limited research on feature selection for this classification problem. Prior studies on thyroid problems categorize them into three classes: normal, hypothyroidism, or hyperthyroidism. However, for proactive prediction and treatment, categorizing patients based on their treatment and general health condition would be more effective. Furthermore, there has been limited discussion on evaluating and comparing the performance of machine learning and deep learning-based techniques for thyroid disease classification. To address these limitations, we propose a multiclass solution for thyroid disease classification that utilizes feature selection and provides a comprehensive performance comparison of machine learning and deep learning-based approaches.

Table 1. An overview of the related research on thyroid disease.

Refs	Year	Sample Size	Dataset	Model	Classes	Results
[40]	2020	-	ToxCast	LR, RF, SVM, XGB, ANN	2	83.00%
[41]	2020	7547	UCI	SVM	3	97.49%
[42]	2021	299	UCI	DT, RF, KNN, SVM, ANN	2	98.50%
[43]	2021	3771	UCI	DT, KNN, RF, ANN	4	96.10% - 98.30%
[44]	2021	7200	UCI	MLP	3	99.00%
[45]	2021	519	DDB	SVM, DT, RF, LR and NB	4	99.35%
[28]	2021	-	UCI	SVM and RF	3	96.80% - 97.30%
[34]	2021	1250	-	SVM, DT, NB, LR, KNN, MLP	3	83.20% - 96.40%
[30]	2022	7200	KEEL & UCI	RF, BME, XGB, AB	3	92.44% - 99.27%
[30]	2021	3010	Kaggle	Ensemble	2	99.60%
[29]	2021	690	KEEL & DHTH	KNN	3	98%
[46]	2022	3152	UCI	DNN	2	99.95%
[47]	2022	3163	UCI	DT, RF, KNN and ANN	2	94.80%
[48]	2022	215	UCI	KNN, XGB, LR, DT	3	81.25% - 87.50%

3. Materials and methods

The methodology employed in this study is depicted in Figure 2 and comprises the following steps: data acquisition, preprocessing, feature importance, class imbalance handling, outlier detection, feature standardization, ensemble modeling and performance evaluation. Two datasets were used for analysis: the KEEL thyroid disease dataset and the hypothyroid dataset from the UCI repository. During the preprocessing phase, various data exploration techniques were applied to gain insights into the datasets. Feature importance techniques were employed to identify the most relevant features for thyroid disease classification. To explore the selected features and their relationships, dimensionality reduction techniques like principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) were employed.

The class imbalance issue was addressed using the adaptive synthetic (ADASYN) sampling technique, which oversampled the minority class to ensure equal representation of all classes. Subsequently, outlier detection techniques were applied to identify and remove anomalous observations from the selected features. The features were then standardized to ensure consistent scaling across all variables. The methodology employed in this study involved the development of an ensemble model composed of two super learners, with each super learner containing three distinct classifiers. This ensemble model aimed to boost the classification performance by leveraging the strengths of multiple classifiers and reducing the overall model variance. The ensemble model was evaluated using a range of performance metrics, such as accuracy, specificity, sensitivity and F1-score, to thoroughly assess its effectiveness in classifying thyroid diseases.

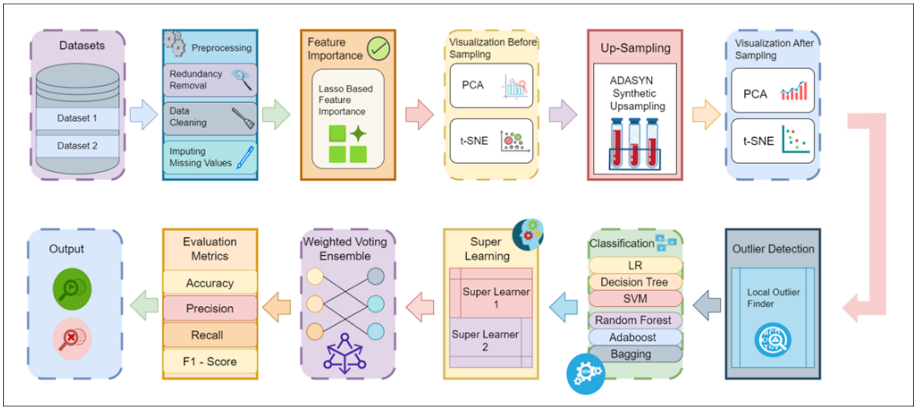


Figure 2. General block diagram of the proposed methodology.

3.1. Dataset description

Two datasets were employed in this work to enhance the analysis and thyroid disease classification. The first dataset, the KEEL thyroid disease dataset, offers a comprehensive collection of attributes related to thyroid function tests, patient demographics and clinical data. The second dataset, the Hypothyroid dataset from the UCI repository, complements the KEEL dataset by providing additional instances and features pertinent to hypothyroidism, a common type of thyroid disorder. By utilizing both datasets, the analysis benefits from a diverse and extensive set of instances that cover a broader spectrum of thyroid disease cases. This comprehensive dataset allows for a more accurate evaluation of the classification models and ensures a robust analysis of the factors influencing thyroid disease classification. The details of each dataset are given as follows:

3.1.1. Dataset 1: KEEL Dataset

The KEEL thyroid disease dataset provides a comprehensive collection of data related to thyroid conditions, enabling us to develop and evaluate machine learning models for diagnosing and predicting thyroid disorders. The dataset combines demographic information (age, sex), medical history (on_thyroxine, on_antithyroid_medication, thyroid_surgery, I131_treatment) and various thyroid-related conditions and treatments (query_on_thyroxine, query_hypothyroid, query_hyperthyroid, lithium, goitre, tumor, hypopituitary, psych) as attributes. It includes essential thyroid hormone levels (TSH, T3, TT4, T4U, FTI), providing valuable insights into the patient' thyroid function. The three classes in the dataset represent distinct thyroid disease categories, enabling researchers to develop multi-class classification models for disease detection and prognosis.

3.1.2. Dataset 2: Hypothyroid Dataset

The second dataset under consideration consists of 30 attributes for 3,772 patients, with 29 variables being categorical and one being an integer value. Dataset-1 has a significant amount of missing data. Among the 30 attributes, eight crucial features contain missing data. These features are TT4, FTI, T4U, age, sex, TSH, T3 and TBG, with 231, 385, 387, 1, 150, 369 and 769 missing samples out of the total 3,772 instances, respectively. The TB feature is entirely comprised of missing values. The target class distribution, represented as a binary class, includes 3,481 samples labeled as P and 291 as N.

3.2. Data preprocessing

In the initial stage of preprocessing, the datasets are thoroughly examined for potential errors and inconsistencies, such as incorrect formatting, duplicate entries and invalid values. These issues are rectified through a meticulous data cleaning process, ensuring the integrity of the data. Subsequently, the datasets are scrutinized for missing values, which are imputed using a variety of techniques, encompassing mean, median, mode and k-nearest neighbor' imputation methods.

In the succeeding step, features that has no significant contribution to the model are identified and eliminated from the datasets. Such features may encompass irrelevant or redundant data or data that exhibits high correlation with other features. This step aids in streamlining the datasets and mitigating noise, which ultimately enhances the model's accuracy and reliability.

Following this, redundant values are identified and removed from the datasets. This process entails detecting and eliminating duplicate data present across the datasets, as well as any additional redundant information that may exist. By eradicating redundant values, the datasets are further simplified, which bolsters the efficiency and accuracy of the machine learning model applied in the study.

3.3. LASSO model-based attribute importance

In conjunction with the pre-processing steps detailed earlier, this study also utilized a least absolute shrinkage and selection operator (LASSO) model-based attribute importance technique to identify significant features from the preprocessed data. The LASSO model, a well-established linear regression model, is frequently employed in machine learning for the purpose of feature selection. The model is advantageous as it not only minimizes the residual sum of squares but also constrains the sum of the absolute values of the coefficients. This constraint leads to the shrinkage of some coefficient estimates to zero, effectively excluding them from the model and resulting in a more parsimonious and interpretable model.

The LASSO model can be represented mathematically as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \;\; + \beta_p x_p + \epsilon \tag{3.1}$$

where y is the dependent variable, $x_1, x_2, , x_p$ are the independent variables, β_0,β_1,β_2, ,β_p are the regression coefficients and ϵ is the error term.

The LASSO model objectively try to minimize the following equation:

$$\frac{1}{2n}(||y - X\beta||)^2 + \lambda||\beta||_1 \tag{3.2}$$

where $(||y - X\beta||)^2$ is the residual sum of squares, λ is the penalty parameter and $||\beta||_1$ is the L1 norm of the coefficients.

The coefficient estimates can be obtained by solving the following equation:

$$\hat{\beta} = \arg\min \frac{1}{2n}(||y - X\beta||)^2 + \lambda||\beta||_1 \tag{3.3}$$

where $\hat{\beta}$ is the coefficient estimates and X is the preprocessed data.

Once the LASSO model was trained on the preprocessed data, we extracted the non-zero coefficients as the most important features for predicting the target variable. These important features were then used as input for the final machine learning model, which was trained and evaluated using standard techniques such as cross-validation and hyperparameter tuning. Figures 3 and 4 illustrate the important features of the first dataset and second dataset using LASSO model, respectively.

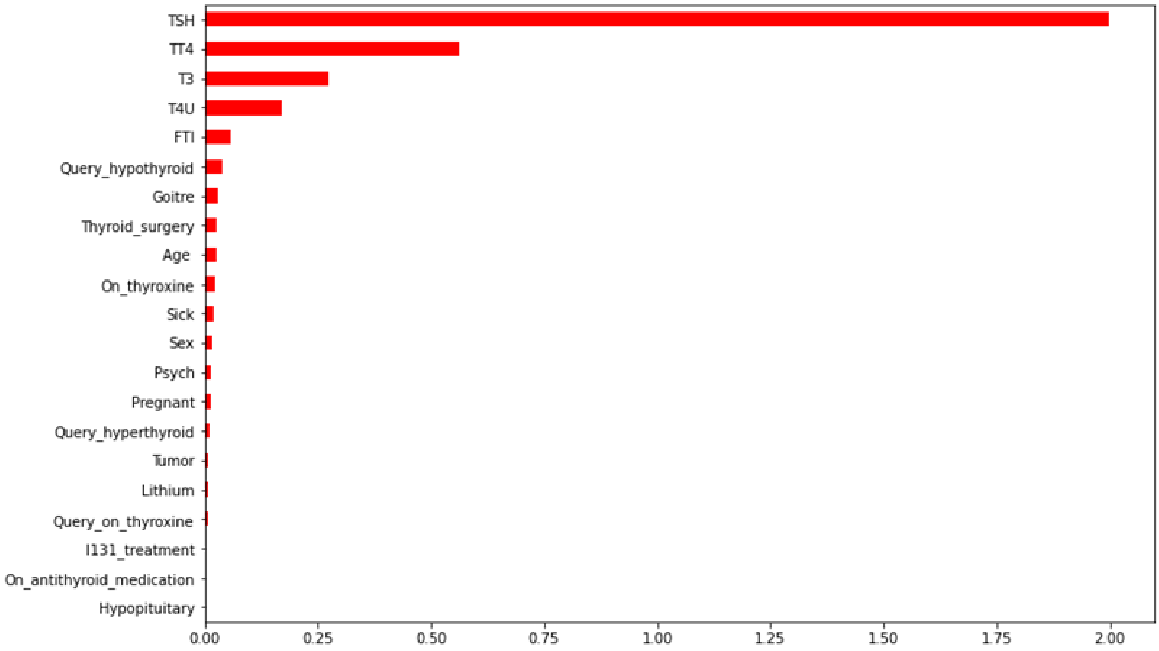


Figure 3. LASSO model feature importance from the first dataset.

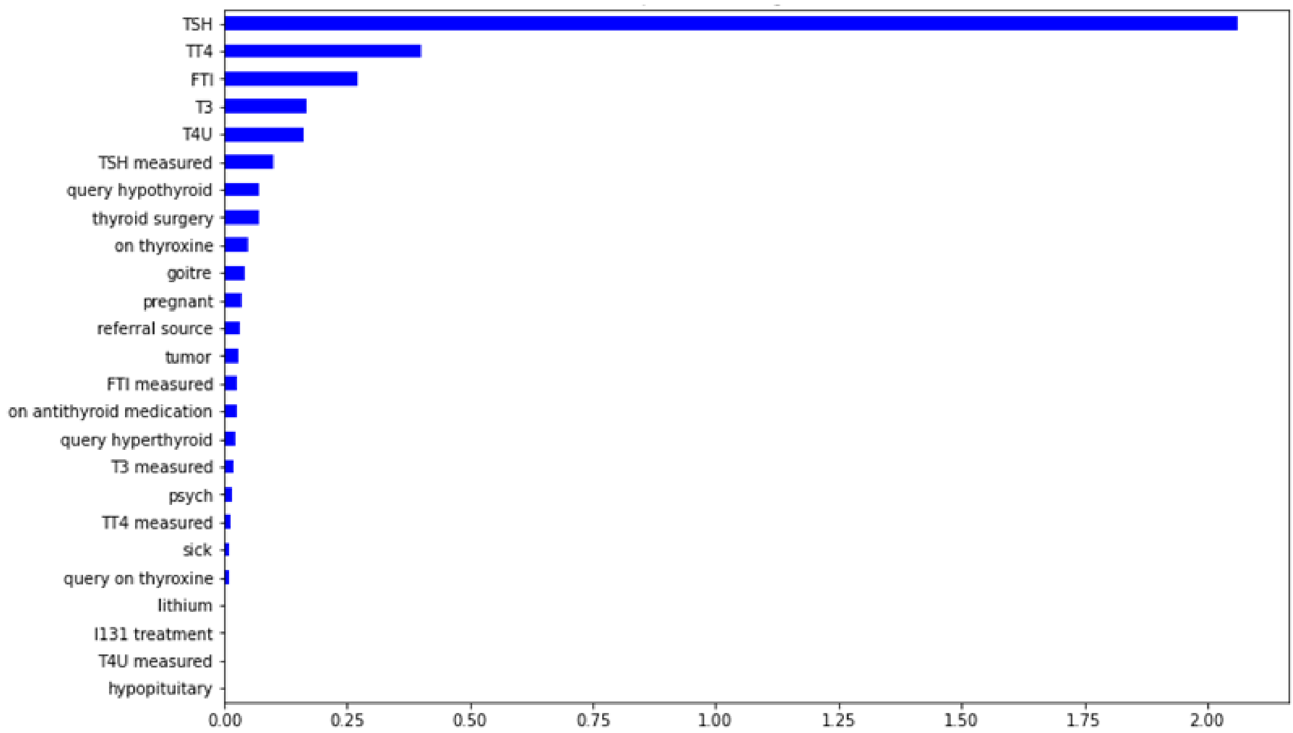


Figure 4. LASSO model feature importance from second dataset.

3.4. Data visualization

Upon determining the most significant features from the preprocessed data using the Lasso model-based attribute importance technique, the subsequent step involves visualizing the data in a manner that emphasizes its inherent structure. In this research, we employed both principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) to investigate the chosen features and discern any patterns or clusters present within the data.

3.4.1. Principal component analysis

Principal component analysis (PCA) is a widely employed dimensionality reduction technique utilized in various fields like image processing, finance and genetics [49]. The method is a mathematical algorithm that endeavors to decrease the number of features within a dataset while preserving the most essential information. PCA does this by transforming the dataset into a new coordinate system that is aligned with the principal components of the original data, where each principal component constitutes a linear combination of the original features. The objective of PCA is to maximize the variance of the data along each principal component, thereby ensuring that the most significant information in the data is retained. PCA is particularly useful for visualizing data in two or three dimensions, but it can also be applied to higher dimensional data.

Given a dataset X, which contains n observations and p features, the first step of PCA is to calculate the covariance matrix C. The covariance matrix describes the relationship between the different features of the dataset. Specifically, it measures how much two features vary together.

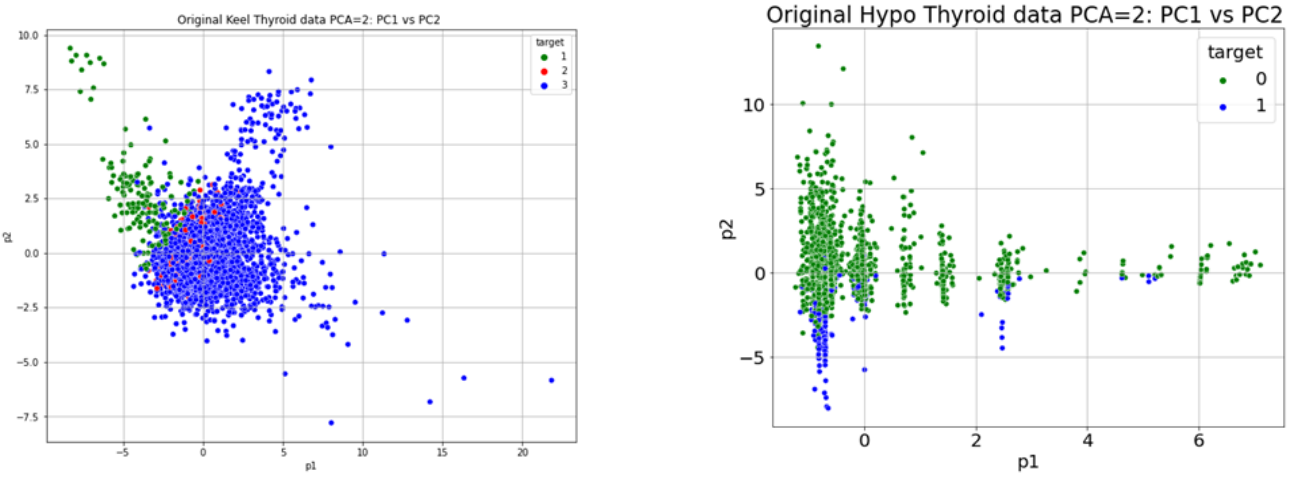
The diagonal elements of the covariance matrix represent the variance of each feature, while the off-diagonal elements represent the covariance between the features.

Next, PCA computes the eigenvectors $v_1, v_2, , v_p$ and the corresponding eigenvalues $\lambda_1, \lambda_2, , \lambda_p$ of the covariance matrix C of X. The eigenvectors $v_1, v_2, , v_p$ form an orthonormal basis for the p-dimensional space and can be used to project the data onto a new coordinate system that captures the maximum amount of variance in the data.

The eigenvectors are the directions in which the data varies the most and the corresponding eigenvalues indicate the amount of variance in the data along these directions. The eigenvectors and eigenvalues are sorted in descending order of the eigenvalues and only the top k eigenvectors are retained. These eigenvectors form an orthonormal basis for the p-dimensional space. PCA then projects the data onto a new coordinate system that captures the maximum amount of variance in the data. The projection of X onto the k-dimensional subspace spanned by the first k eigenvectors is given by the matrix multiplication:

$$Z = X \times V_k \tag{3.4}$$

where V_k is the matrix consisting of the first k eigenvectors of C. The projected data Z has dimensions $n \times k$, where k is the number of retained eigenvectors. The resulting projected data Z can be used for further analysis or visualization. PCA is particularly useful when dealing with high-dimensional datasets, as it can significantly reduce the number of features while retaining the most important information. PCA is also used for feature extraction, anomaly detection and clustering. Figure 5 (a) presents the projection of PCA of first dataset, while Figure 5 (b) presents the PCA projection of second dataset before resampling.



(a)

(b)

Figure 5. Visualization of attributes in original datasets using PCA (before resampling): (a) first dataset; (b) second dataset.

3.4.2. t-Distributed stochastic neighbor embedding

t-Distributed stochastic neighbor embedding (t-SNE) is a commonly utilized technique for nonlinear dimensionality reduction which enables the representation of high-dimensional data in a lower-dimensional space in a visual format.. Given a dataset X, which contains n observations and p features, t-SNE constructs a lower-dimensional map Y where the distances between points reflect the similarities in their probabilities.

The first step of t-SNE is to model the high-dimensional data as a set of probabilities. Specifically, it constructs a probability distribution P over pairs of high-dimensional data points such that similar points have a higher probability of being chosen than dissimilar points. It then constructs a probability distribution Q over pairs of low-dimensional data points that aims to preserve the similarity structure of the high-dimensional data. The algorithm works by minimizing the Kullback-Leibler divergence between the joint probabilities P and the conditional probabilities Q.

The cost function to be minimized is given by $C = KL(P||Q)$, where KL is the Kullback-Leibler divergence. This cost function measures the difference between the probability distributions P and Q. The cost function to be minimized is given by:

$$C = KL(P||Q) = \sum_i \sum_j P_{ij} \log\left(\frac{P_{ij}}{Q_{ij}}\right) \tag{3.5}$$

The probability P_{ij} that point i would choose point j as its neighbour in the high-dimensional space is computed using a Gaussian kernel:

$$P_{ij} = \frac{\exp\left(\frac{-||x_i - x_j||^2}{2\sigma_i^2}\right)}{\sum_{k\neq l} \exp\left(\frac{-||x_k - x_l||^2}{2\sigma_k^2}\right)} \tag{3.6}$$

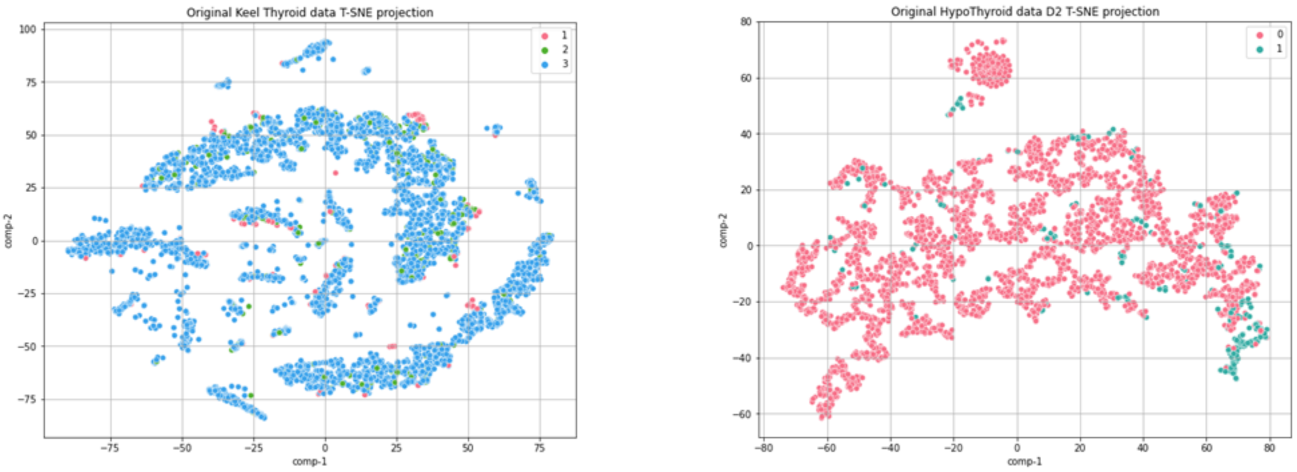
where x_i and x_j are the feature vectors of points i and j in the high-dimensional space and $||x_i - x_j||$ is the Euclidean distance between them. The parameter σ_i is the standard deviation of the Gaussian kernel for point i and is computed as the distance to its kth nearest neighbor. This parameter is chosen to reflect the density of the data around each point, which helps to balance the probabilities for points in dense and sparse regions of the data.

To compute the probability Q_{ij} that point i would choose point j as its neighbor in the low-dimensional space, t-SNE uses the students t-distribution. Specifically, it defines Q_{ij} as:

$$Q_{ij} = \frac{\left(1 + ||y_i - y_j||^2\right)^{-1}}{\sum_{k\neq i}\left(1 + ||y_i - y_k||^2\right)^{-1}} \tag{3.7}$$

where y_i and y_j are the coordinates of points i and j in the low-dimensional space and $||y_i - y_j||$ is the Euclidean distance between them. To normalize the probabilities, the parameter in the denominator represents the summation over all other points in the low-dimensional space.

By employing gradient descent, t-SNE reduces the cost function with respect to the coordinates of the points in the low-dimensional space, y_i. This is accomplished by continuously updating the coordinates of the points until the cost function reaches a minimum. The algorithm is recognized for its capacity to preserve the local structure of data, which renders it highly advantageous for visualizing intricate datasets like images and text. Figure 6 shows the t-SNE projections of first and second dataset, respectively.



(a)

(b)

Figure 6. Visualization of attributes in original datasets using t-SNE (before resampling): (a) first dataset, (b) second dataset.

3.5. Adaptive sampling

The issue of class imbalance in machine learning has long been recognized as a significant challenge, as it can introduce bias in favor of one class while under-representing another. To address this challenge, a range of sampling techniques have been developed over time, including synthetic over-sampling, which involves generating artificial data points to represent the minority class.

Among the various synthetic sampling methods, ADAptive SYNthetic (ADASYN) sampling has emerged as particularly effective, owing to its non-linear interpolation scheme. This approach introduces non-linearity to the sampled dataset by generating synthetic examples that lie between existing minority examples and their k-nearest neighbors from the majority class. These synthetic samples are then generated in accordance with the density distribution of the minority class in the feature space, which captures the underlying non-linear relationship between the minority and majority classes.

As a result, ADASYN generates minority samples that are uniquely representative of the minority group, introducing new patterns and variations within the dataset. This, in turn, can enhance the ability of machine learning models to capture the non-linear relationships between the feature and target variables.

Let X be a dataset with N samples and M features. Let C_1 be the majority class and C_2 be the minority class, where $C_1 < C_2$. To reduce the class imbalance, we use ADASYN sampling to obtain $C_1 = C_2 \times \beta$, where $\beta \in [0, 1]$ is the desired sampling level.

To apply ADASYN sampling, we first calculate the density distribution of minority class samples. For each minority sample x_i in class C_2, the density distribution $D(x_i)$ is calculated as:

$$D(x_i) = \sum_{j=1}^{N} w_j(x_i) \times \frac{1}{dist(x_i, x_j)^p} \tag{3.8}$$

where $D(x_i)$ is the density distribution of the ith minority sample $dist(x_i, x_j)$ is the Euclidean distance between the ith and jth nearest neighbor samples from both classes C_1 and C_2 and p is the decay parameter that controls the rate of decay of contribution of distant samples to the density distribution. The weight $w_j(x_i)$ is a function that assigns a weight to each sample based on its similarity to x_i:

$$w_j(x_i) = \exp\left(\frac{-d_j(x_i)^2}{2 * \sigma^2}\right) \tag{3.9}$$

where $d_j(x_i)$ is the Euclidean distance between x_i and x_j and σ is a bandwidth parameter.

The class imbalance ratio I_r is determined by calculating the ratio between the number of majority class samples S_M and the number of minority class samples S_m:

$$I_r = \frac{S_M}{S_m} \tag{3.10}$$

To determine the number of synthetic samples to generate for each minority sample x_i, we use the following equation:

$$G(x_i) = round(D(x_i) \times I_r \times (1 - \alpha)) \tag{3.11}$$

where $G(x_i)$ is the generated synthetic sample for the ith minority sample, α is a hyperparameter that controls the degree of randomness in the sampling process.

Finally, the synthetic samples are generated in a series of iterations. For each minority sample x_i, we select k_i nearest neighbors from both classes C_1 and C_2, where k_i is a hyperparameter. We then use the following mathematical equation to generate the k_i synthetic samples:

$$SS_k = x_i + \alpha_k \times (x_r - x_i) + \beta_k \times (x_j - x_i) \tag{3.12}$$

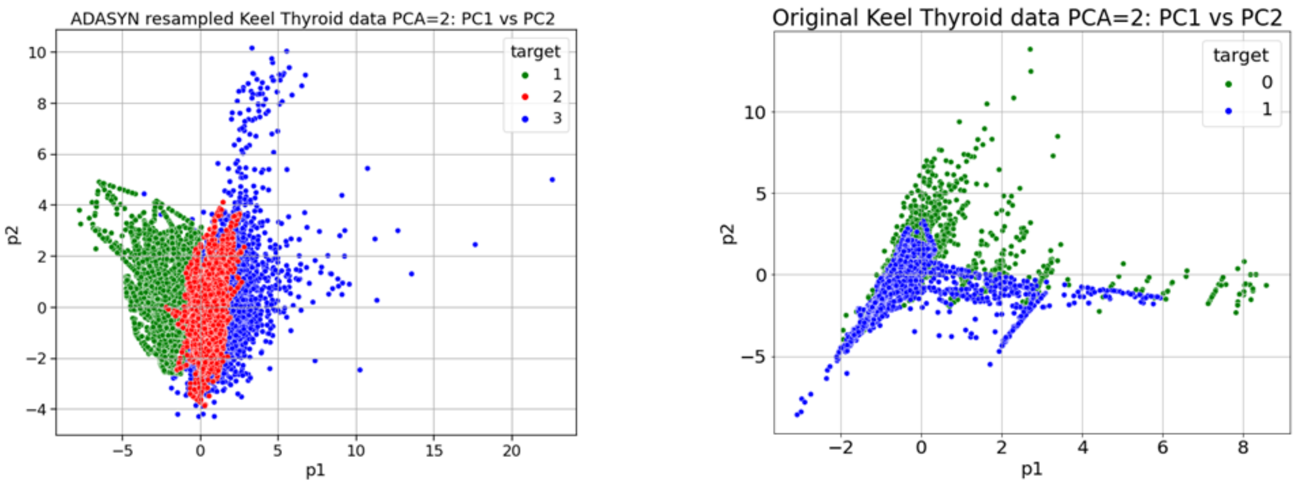
where SS_k is the kth synthetic sample generated for the ith minority sample, α_k and β_k are random numbers between 0 and 1, x_r is a randomly chosen minority sample and x_j is a randomly chosen sample from the k_i nearest neighbors.

The values of α_k and β_k are determined using the following relations:

$$\alpha_k = (1 - \delta) * r_1 + \frac{\delta}{2} \tag{3.13}$$

$$\beta_k = (1 - \delta) * r_2 + \frac{\delta}{2} \tag{3.14}$$

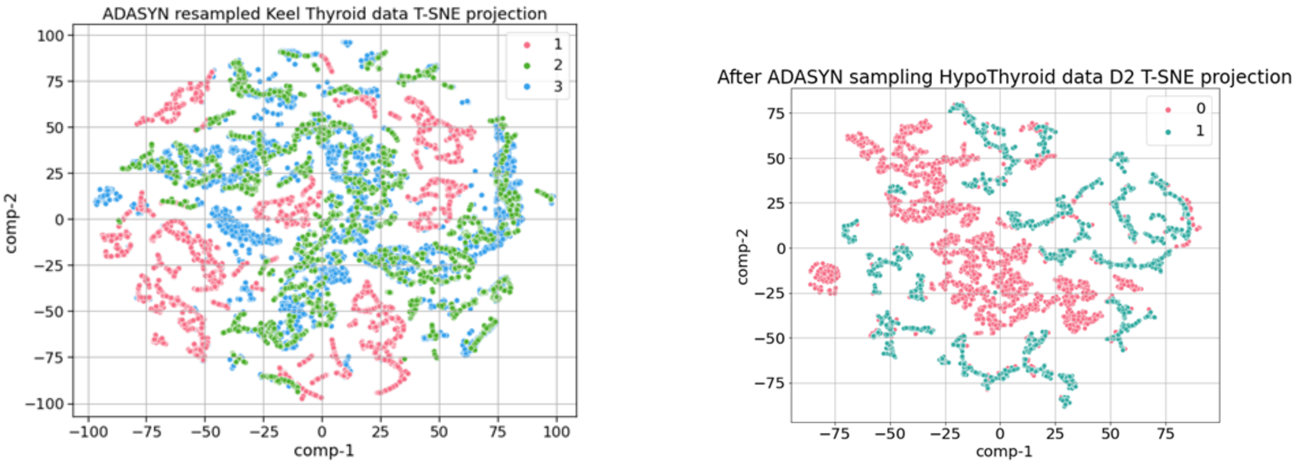
where δ is a hyperparameter that controls the degree of randomness in the sampling process and r_1 and r_2 are random numbers between 0 and 1. Figure 7 shows the target variable distribution of first dataset and second dataset after resampling using ADASYN technique with PCA while Figure 8 shows the target variable distribution of first dataset and second dataset after resampling using ADASYN technique with t-SNE.



(a)

(b)

Figure 7. Visualization of both datasets using PCA after ADASYN resampling: (a) first dataset after resampling; (b) second dataset after resampling.

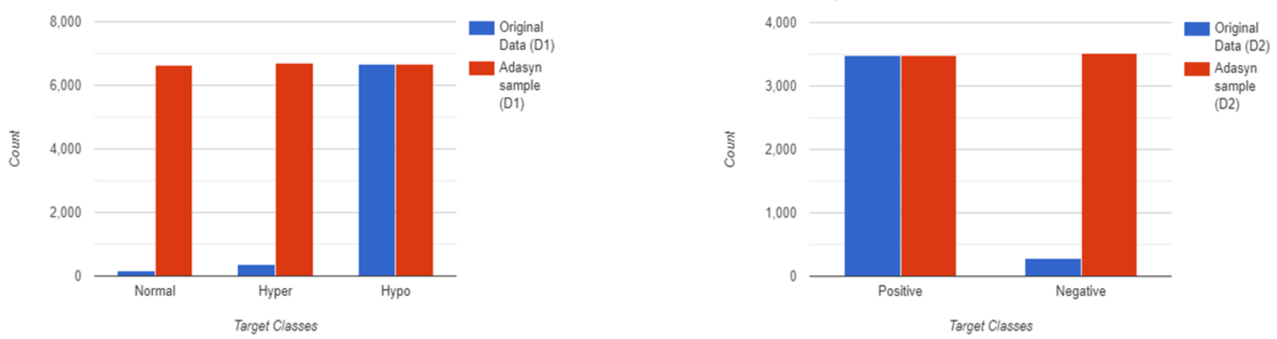


(a)

(b)

Figure 8. Visualization of both datasets using t-SNE after ADASYN resampling: (a) first dataset after resampling; (b) second dataset after resampling

Figure 9 illustrates the comparison of the target variable distribution between the original and resampled datasets for the two datasets. Figure 9 (a) represents the first dataset before and after applying ADASYN resampling and Figure 9 (b) represents the second dataset before and after applying ADASYN resampling.



(a) (b)

Figure 9. Comparison of target variable distribution of original and resampled datasets: (a) first dataset before and after ADASYN resampling; (b) second dataset before and after ADASYN resampling.

It can be observed that the distribution of both classes has become more balanced in the resampled datasets. This indicates that the ADASYN resampling technique has successfully addressed the class imbalance problem in both datasets, which can potentially lead to more accurate and robust classification models.

3.6. Local Outlier Finder

The Local Outlier Factor (*LOF*) algorithm is a popular technique for detecting anomalies in a dataset. In this research, we employed the LOF algorithm as part of our methodology for identifying outliers in our dataset. To implement this algorithm, we first defined a distance metric between data points using the commonly used Euclidean distance. We then used the algorithm to calculate the local density of each data point by measuring the average distance between the point and its k-nearest neighbors. To determine the value of k, we performed a sensitivity analysis and chose the value that resulted in the best performance.

With the local density of each data point calculated, we proceeded to compute the LOF score for each point. This score reflects the degree to which a data point deviates from its neighbors in terms of local density. Specifically, a data point with an LOF score significantly lower than its neighbors is considered an outlier.

Let $X = x_1, x_2, , x_n$ be a dataset consisting of n data points, where each data point x_i belongs to a d-dimensional feature space. We define the distance metric between two data points x_i and x_j as the Euclidean distance, given by the equation $dist(x_i, x_j) = \sqrt{(x_i x_j)^T (x_i x_j)}$, where T denotes the transpose operator.

Given a data point x_i, we define its k-distance as the distance between x_i and its kth-nearest neighbor, given by the equation:

$$k - distance(x_i) = dist(x_i, x_k(i)) \tag{3.15}$$

where $x_k(i)$ is the kth-nearest neighbor of x_i. Using the k-distance of each data point, we define the local reachability density (*LRD*) of a data point x_i as the inverse of the average k-distance of x_i's k-nearest neighbors. This is given by the equation:

$$LRD(x_i) = \left(\frac{\sum_{x_i,x_j} dist(x_i, x_j)}{k} \right)^{-1} \tag{3.16}$$

where the sum is taken over x_i's k-nearest neighbors.

Finally, we define the LOF score of a data point x_i as the average ratio of the *LRD* of x_i to the *LRD* of its k-nearest neighbors, given by the equation:

$$LOF(x_i) = \frac{\sum \left(\frac{LRD(x_j)}{LRD(x_i)} \right) /k}{\left(\sum LRD(x_j) \right) /k} \tag{3.17}$$

where the sum is taken over x_i's k-nearest neighbors, excluding x_i itself.

Table 2 shows the improvement in the number of training samples for both datasets before and after applying ADASYN and outlier removal. Dataset 1 had 5,760 samples in the original dataset, which increased to 14,410 after applying ADASYN, and reduced to 12,969 after outlier removal. Similarly, Dataset 2 had 3,017 samples in the original dataset, which increased to 5,592 after applying ADASYN and then reduced to 5,032 after outlier removal. The number of samples increased significantly after applying ADASYN, which helps to balance the class distribution in both datasets.

Table 2. Improvement of number of training samples in both dataset; before and after utilization of ADASYN and outlier removal.

	Total number of samples in original dataset	**Outlier detection**	**After outlier removals**	**Total training samples after ADASYN**	**Outlier detection after ADASYN**	**Total number of samples after outlier removal**
Dataset 1	5760	576	5184	14410	1441	12969
Dataset 2	3017	302	2715	5592	560	5032

3.7. Classification models

In this research study, we utilized two iterations of the super learner (SL) ensemble technique as our primary methodology to predict outcomes in our dataset. The SL ensemble technique is an effective method for combining multiple machine learning models to achieve higher prediction accuracy.

In our study, we employed two SL ensembles that consisted of three base estimators each. The first SL ensemble comprised of logistic regression, decision trees and support vector classification. We selected these estimators based on their individual strengths and potential synergies when combined. To combine the predictions of the base estimators, we utilized a random forest as the meta estimator known for its ability to reduce overfitting and improve prediction accuracy. Similarly, for the second SL ensemble, we selected random forest, adaBoost and bagging Classifier as base estimators based on their respective strengths in handling large datasets, improving weak learners' performance and reducing overfitting. In this study, a decision tree was employed as the meta estimator to integrate the predictions of individual models. This was achieved by recursively partitioning the dataset into smaller subsets based on input features, until a stopping criterion was reached. The decision tree algorithm then predicted the output variable based on the most prevalent class within each subset and these predictions were used to generate the final prediction. Figure 10 shows the overall overview of implemented model in this study.

By utilizing two iterations of the SL ensemble technique with different combinations of base estimators and meta estimators, we were able to achieve higher prediction accuracy than any individual model could achieve alone. After applying SL 1 and SL 2 ensemble techniques to our dataset, we wanted to further improve the accuracy of our predictions. Therefore, we decided to use a weighted average voting technique as our final step.

After calculating the performance metrics for each model, we assigned weights to each model based on their performance. We assigned higher weights to the models with better performance and lower weights to those with weaker performance. The weights were assigned in such a way that the total sum of weights was equal to one. After assigning the appropriate weights to each model, we combined their predictions by calculating a weighted average of their outputs. To obtain the final prediction, we took a weighted average of the predicted probabilities for each potential outcome. This approach allowed us to derive a more accurate prediction by taking into account the strengths and weaknesses of each individual model.

To gauge the effectiveness of the weighted average voting technique, we compared its performance to that of the individual models and super learner ensembles. We utilized several performance metrics, including accuracy, precision, recall and F1-score, to evaluate the effectiveness of the weighted average voting technique. Further details on the classifiers utilized in this study are provided in the subsequent subsections.

3.7.1. Classifier 1: Logistic regression

Logistic regression is a popular and powerful algorithm for supervised machine learning that can be used for binary classification tasks [40]. The goal of logistic regression is to estimate the probability of a binary outcome based on one or more input features [34]. The input features are combined with a set of weights and an intercept term to produce a linear combination of the inputs. This linear combination is then passed through a sigmoid function to obtain the predicted probability.

The logistic regression algorithm determines the weight and intercept terms that minimize the disparity between the predicted probabilities and the factual binary outcomes within the training data. This is accomplished by reducing a cost function utilizing an optimization algorithm like gradient descent. The logistic regression model can be expressed as follows:

$$p(y = 1|x) = \frac{1}{1 + \exp(-(w^T x + b))} \tag{3.18}$$

where $(y = 1|x)$ is the predicted probability of $y = 1$ given input features x, w is the weight vector and b is the intercept term.

To train the logistic regression model, we use a training set of input features and binary target variables. The weights and intercept term are initialized randomly and the cost function is iteratively minimized using an optimization algorithm such as gradient descent. The resulting model can then be used to predict the binary outcome for new input features, as Figure 10.

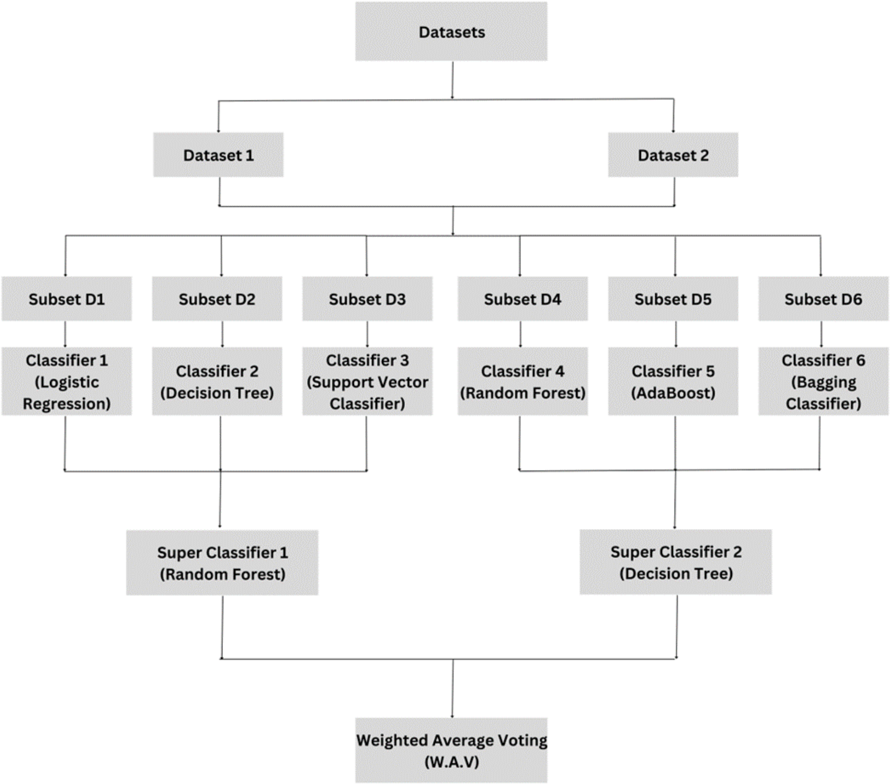

Datasets
Dataset 1
Dataset 2
Subset D1
Subset D2
Subset D3
Subset D4
Subset D5
Subset D6
Classifier 1 (Logistic Regression)
Classifier 2 (Decision Tree)
Classifier 3 (Support Vector Classifier)
Classifier 4 (Random Forest)
Classifier 5 (AdaBoost)
Classifier 6 (Bagging Classifier)
Super Classifier 1 (Random Forest)
Super Classifier 2 (Decision Tree)
Weighted Average Voting (W.A.V)

Figure 10. The overview of implemented classification models in this study.

3.7.2. Classifier 2: Decision trees

The decision trees algorithm is a highly adaptable supervised machine learning model that can accommodate both categorical and numerical data and perform both classification and regression tasks [29]. The algorithm does this by recursively dividing the data into smaller subsets based on the most important attributes until a stopping criterion is reached [34]. The resulting structure is a visual representation of a decision-making process, with nodes representing decisions based on particular attributes and branches representing the outcomes of those decisions [48]. The root node signifies the initial decision with maximum entropy, while the leaf/terminal nodes indicate the final decisions with zero entropy [50]. This approach has proven highly effective in a wide range of problem domains and can offer valuable insights into complex decision-making processes.

Assuming we have a dataset $D_1 = \{(x_1, y_1), (x_2, y_2), (x_n, y_n)\}$ with input features $X = \{x_1, x_2, x_3, x_4, , x_n\}$ and a target variable Y. where,

- D_1 is the dataset with x_i input features and y_i target variables.
- $x_i = \{x_i1, x_i2, x_i3, ., x_in\}$ is the feature vector for ith observation.
- $y_i = \{y_i1, y_i2, y_i3, ., y_in\}$ is the ith target variable.

To select the root feature, we use an entropy-based impurity measure. Entropy is calculated using the following:

$$H(s) = -\sum_{x} p(x) \log p(x) \tag{3.19}$$

where, $H(s)$ is the entropy and $p(x)$ is the percentage of class x in the attribute node.

The Information Gain after the split is calculated using the following:

$$G(S, x_{in}) = H(s) - \sum_{v \in Values(x_{in})} \frac{|S_v|}{|S|} H(s_v) \tag{3.20}$$

where,

- $G(S, x_{in})$ is the gain of nth feature of [i]th observation,
- x_{in} is the nth feature within the root node,
- S is the class subset of $x_{[}in]$ feature,
- $H(s)$ is the entropy of the root,
- $H(s_v)$ is the entropy of the child nodes,
- s_v are the samples in respective subset,
- s are the samples in the root node.

The Gain for each feature in x_{in} is computed and the feature that maximizes the impurity reduction is selected, i.e., $G(S, x_{i1}) > G(S, x_{i2})$. This process is iterated until a stopping criterion is met, such as reaching the maximum depth or minimum impurity reduction threshold or having a minimum number of observations per node.

3.7.3. Classifier 3: Support Vector Classification

Support vector classification (SVC) or support vector machines (SVM) is a popular and effective supervised machine learning algorithm that can be used for both classification and regression tasks [34]. The goal of SVC is to find a hyperplane that best separates the input data into different classes or to find a hyperplane that best fits the input data for regression tasks [45, 51]. The hyperplane is chosen to maximize the margin, or the distance between the hyperplane and the closest data points from each class [40, 41].

To train an SVC model, we first select a kernel function, denoted as $K(x, x')$, that maps the input features to a higher-dimensional space, where the input data is more separable [50]. The kernel function takes two input vectors x and x' and outputs a scalar value that measures the similarity between them. The most common kernel functions are linear, polynomial and radial basis function (RBF) [51]. The choice of kernel function depends on the characteristics of the input data and the specific problem domain [52].

The input data consists of n feature vectors x_i, where $i \in [1, n]$ and the corresponding binary labels y_i, where $y_i \in \{-1, 1\}$ for classification tasks and $y_i \in \Re$ for regression tasks. For classification tasks, we aim to find a hyperplane in the feature space that separates the two classes with the largest possible margin. For regression tasks, we aim to find a hyperplane that best fits the input data with minimum error.

The optimization problem for SVM is defined as follows:

$$minimize\left(\frac{1}{2}\|w\|^2 + C\sum_i \xi_i\right) \tag{3.21}$$

subject to $y_i(w^T x_i + b) \geq 1 - \psi_i$ and $\xi_i \geq 0$ where w is the weight vector, b is the bias term and ξ_i is the slack variable that allows for misclassifications in the margin. The parameter C controls the trade-off between maximizing the margin and minimizing the classification or regression error [52].

The solution to the optimization problem is obtained by solving its dual form, which is given by:

$$\max \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{3.22}$$

subject to $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i y_i = 0$ where α_i is the Lagrange multiplier associated with the ith data point and the support vectors are the data points with non-zero Lagrange multipliers. The weight vector w and the bias term b can be computed from the support vectors and their corresponding Lagrange multipliers.

Once the hyperplane is determined, new input data can be classified or predicted by computing its distance from the hyperplane. For classification tasks, the predicted label is determined by the sign of the distance.

3.7.4. Classifier 4: Random forest

The algorithm is an influential machine learning technique that finds extensive application in both classification and regression tasks [53]. It falls within the category of ensemble learning algorithms, which entails the combination of multiple decision trees to produce more accurate predictions [47]. One of the principal advantages of random forest is its utilization of bootstrap aggregation (also referred to as bagging) to enhance the performance of the model by decreasing variance [50]. This is achieved by training each decision tree on a randomly selected subset of the original data, which helps to mitigate overfitting and enhance the generalization capability of the model.

For a given dataset X consisting of n examples, where each example has m features and Y is the respective set of class labels, the random forest algorithm aims to learn a function $f : X \rightarrow Y$ that can predict the class for a new input vector x.

To create a random forest classifier, we first generate a set of D_i bootstrap samples of size n' that are uniformly and randomly selected with replacement from the original dataset X. Since observations are selected with replacement, there may exist some duplicates within each D_i. A fraction $f''(1 - \frac{1}{e} \approx 63.2\%)$ of the unique examples may exist for $n' = n$ in D_i, while the remaining examples are duplicates.

We can represent X as a collection of bootstrap samples, D_i, where:

$$X = D_i = \{D_1, D_2, D_3, ..., D_n\} \tag{3.23}$$

$$D_i = \sum_{i=0}^{n} D_i \tag{3.24}$$

$$\vDash D_i, \begin{cases} f' = 63.2\%, n' = n \\ f'' < 63.2\%, n \neq n \end{cases} \tag{3.25}$$

For each feature F and target variable T in each sample set D_i, we calculate entropy and gain to create $n = D_i$ decision trees. The entropy of a selectable feature F and the target variable T is calculated using:

$$E(T, F) = \sum_{c \in F} P(c)E(c) \tag{3.26}$$

where $E(c)$ is the entropy of the respective class and $P(c)$ is the proportion of samples belonging to the respective class.

We can then calculate the information gain after the split using:

$$Gain(T, F) = E(T) - E(T, F) \tag{3.27}$$

where $E(T)$ is the entropy of the target variable and $E(T, F)$ is the entropy of the target and the feature.

The predicted outcome t from each of the bootstrap samples in D_i is then compared to form an aggregate score:

$$\hat{y} = \arg\max_i \left(\sum_{j=1}^{n} \delta(\hat{y}_j) = t \right), \ t \in \{0, 1\} \tag{3.28}$$

where $\hat{y}$ is the predicted outcome and δ is the Kronecker delta function.

Finally, the predictions from a random forest algorithm can be given by:

$$f(x) = \frac{1}{N} \sum_{i}^{N} T_i(x) \tag{3.29}$$

where $f(x)$ is the predicted class label for an input vector x, N is the number of decision trees in the forest and $T_i(x)$ is the prediction of the ith decision tree. The scaling factor $\frac{1}{N}$ ensures that the output is a probability distribution over possible class label.

3.7.5. Classifier 5: AdaBoost

AdaBoost, short for adaptive boosting, is a popular ensemble learning algorithm used for binary classification and regression tasks [30]. AdaBoost combines multiple weak classifiers into a strong classifier by assigning weights to each weak classifier based on their accuracy [53]. Let C(x) be the binary classifier that predicts the label of input data x. The final prediction of the Adaboost model is given by:

$$H(x) = sign\left(\sum \alpha_t * C_{t(x)}\right) \tag{3.30}$$

where $H(x)$ is the final prediction, α_t is the weight assigned to weak classifier C_t and $sign$ is the sign function that returns $+1$ or -1 depending on the sign of its argument.

To update the weights of the input data samples after each iteration, we use the following formula:

$$w_i = w_i * \exp(-\alpha_t * y_i * C_{t(x_i)}) \tag{3.31}$$

where w_i is the weight of data point i, y_i is the true label of data point i and x_i is the input data. If data point i is correctly classified by weak classifier C_t, $y_i * C_{t(x_i)}$ is positive and the weight w_i is decreased. If data point i is misclassified, $y_i * C_{t(x_i)}$ is negative and the weight w_i is increased.

The weight assigned to each weak classifier is determined by its accuracy on the training data. Let ϵ_t be the classification error of weak classifier C_t, defined as:

$$\eta_t = \frac{\sum(w_i * |y_i - C_{t(x_i)}|)}{\sum(w_i)} \tag{3.32}$$

The weight α_t is then computed as:

$$\alpha_t = 0.5 * \ln \frac{1 - \eta_t}{\eta_t} \tag{3.33}$$

The weight α_t is positive if the classification error of C_t is less than 0.5 and negative otherwise. A higher weight is assigned to weak classifiers with lower classification error.

3.7.6. Classifier 6: Bagging classifier

The bagging classifier is a powerful ensemble learning method that utilizes multiple independently trained classifiers to enhance prediction accuracy and mitigate overfitting [47]. Bagging Classifier, a contraction of bootstrap aggregating, generates several bootstrap samples of the input data and trains individual classifiers on each sample [50].

The bagging classifier training process commences by randomly selecting data points from the input dataset with replacement, creating several bootstrap samples. Subsequently, a classifier is trained on each bootstrap sample, utilizing the same learning algorithm. Upon completing the training phase, the trained classifiers are employed to make predictions on unseen data. The final prediction is derived by consolidating the predictions of each classifier via majority voting. The bagging classifier's final prediction for a given input sample can be represented as:

$$\hat{y} = mode(\hat{y}_1, \hat{y}_2, \hat{y}_3, ..., \hat{y}_T) \tag{3.34}$$

where $\hat{y}$ is the final prediction of the bagging classifier, $\hat{y}_1, \hat{y}_2, \hat{y}_3, ..., \hat{y}_T$ are the predictions of individual classifiers and mode is the statistical mode, which is the value that appears most frequently in the set of predictions.

3.7.7. Super learners

Super learners (SL) are ensemble methods that combine multiple machine learning models to improve prediction accuracy and reduce overfitting. The SL algorithm uses two stages: base learning and meta learning. During the base learning stage, multiple models are trained using the training data, while in the meta learning stage, a meta model is used to combine the predictions of these base models to generate the ultimate prediction. Figure 11 illustrates the implementation of super learner in this study. Let X be the input data, y be the target variable and M be the set of base models. For each base model m in M, let $f_m(X)$ be the predicted outcome of m on X. Then, the super learner output is given by:

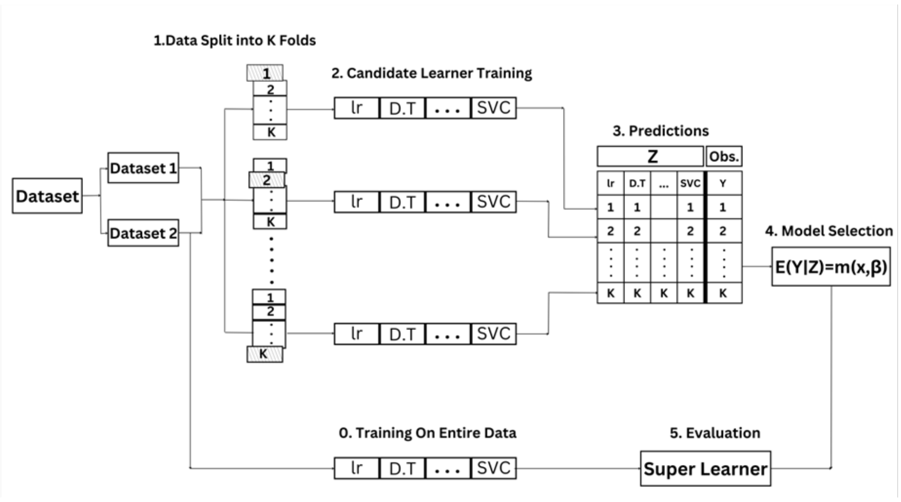


Figure 11. The implementation of super learner in this study.

$$SL(X) = g(f_1(X), f_2(X), ..., f_m(X)) \tag{3.35}$$

where g is the meta model that combines the predictions of the base models.

For our first SL ensemble, we selected three base estimators: logistic regression (LR), decision trees (DT) and support vector classification (SVC). We chose these estimators based on their individual strengths and potential synergies that could be achieved by combining them. Once we had trained our base estimators, we used a random forest as the meta estimator to combine their predictions. The random forest algorithm is known for its ability to reduce overfitting and improve prediction accuracy by using multiple decision trees.

For our second SL ensemble, we selected three different base estimators: random forest (RF), adaBoost and bagging classifier. The random forest algorithm was selected due to its capacity to handle large datasets with high dimensionality, the AdaBoost algorithm was chosen for its effectiveness in enhancing the performance of weak learners and the bagging classifier was chosen for its ability to alleviate overfitting and enhance generalization. We trained and evaluated each of these base estimators using various metrics to identify their strengths and weaknesses. To combine the predictions of the base estimators in our second SL ensemble, we used a DT as the meta estimator. The decision tree algorithm is a simple yet powerful algorithm that recursively splits the dataset into smaller subsets based on input features to predict the outcome.

3.7.8. Weighted average voting

In our research, we also employed weighted average voting (WAV) as another ensemble method to combine the predictions of our base estimators. The weights assigned to each base estimator were based on their performance on the training data. The weights assigned to each base estimator depend on their performance on the training data. The better a base estimator performs on the training data, the higher its weight in the ensemble. The weighted average of the predicted values of the base estimators can be represented as:

$$\hat{y}_{WAV} = \sum_{i=1}^{n} w_i \hat{y}_i \tag{3.36}$$

where $\hat{y}_{WAV}$ is the final prediction of the ensemble, n is the number of base estimators, $\hat{y}_i$ is the predicted value of the ith base estimator and w_i is the weight assigned to the ith base estimator. We found that WAV can be a simple yet effective ensemble method for combining the predictions of multiple base estimators, especially when the base estimators have comparable performance on the training data.

3.8. Performance evaluation metrics

In this section, we will provide a brief overview of utilized performance evaluation metrics, along with our model.

3.8.1. Accuracy

Accuracy is one of the most used performance evaluation metrics and it measures the proportion of correct predictions made by a model. Mathematically, accuracy is defined as follows:

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{3.37}$$

3.8.2. Precision

Precision is a performance evaluation metric that measures the proportion of true positives (i.e., correct positive predictions) out of all positive predictions made by a model. Mathematically, precision is defined as follows:

$$Precision = \frac{\text{Number of true positives}}{\text{Number of true positives + Number of false positives}} \tag{3.38}$$

3.8.3. Recall

Recall is a performance evaluation metric that measures the proportion of true positives (i.e., correct positive predictions) out of all actual positive instances in the dataset. Mathematically, recall is defined as follows:

$$Recall = \frac{\text{Number of true positives}}{\text{Number of true positives + Number of false negatives}} \tag{3.39}$$

3.8.4. F1-Score

The F1 score is a performance evaluation metric that combines precision and recall providing a single metric that balances both metrics. The F1 score is defined as the harmonic mean of precision and recall and is calculated as follows:

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.40}$$

4. Results

In this research study, we utilized two iterations of the super learner (SL) ensemble technique as our primary methodology for predicting outcomes in our dataset. The SL ensemble technique is an effective method for combining multiple machine learning models to achieve higher prediction accuracy. We used an 80-20 data splitting approach to train and test our models. Two super learner (SL) ensembles with three base estimators each were employed to enhance prediction accuracy. The first SL ensemble included logistic regression, decision trees and support vector classification, while the second SL ensemble comprised random forest, AdaBoost andbBagging classifier. To improve prediction accuracy further, a weighted average voting technique was utilized based on the performance of the individual models. After evaluating the performance of the weighted average voting technique using various metrics such as accuracy, precision, recall and F1-score, we found that the ensemble model consistently outperformed individual models. The results highlight the effectiveness of our methodology in predicting thyroid disease outcomes with high accuracy, highlighting the benefits of using SL ensembles and the weighted average voting technique. We assessed the effectiveness of the weighted average voting technique by comparing its performance to that of the individual models and super learner ensembles. We employed multiple performance metrics, including accuracy, precision, recall and F1-score, to evaluate the performance of the weighted average voting technique.

Table 3 shows the performance of the proposed methodology for the first dataset. The original dataset without ADASYN resampling had an accuracy of 99.58%, precision of 99.48%, recall of 95.87% and F1-score of 97.56%. After resampling without ADASYN, the accuracy improved to 99.90%, precision to 99.89%, recall to 99.90% and F1-score to 99.90%. This improvement in performance indicates that the proposed methodology is effective in dealing with imbalanced datasets.

Table 3. Performance of proposed methodology for first dataset.

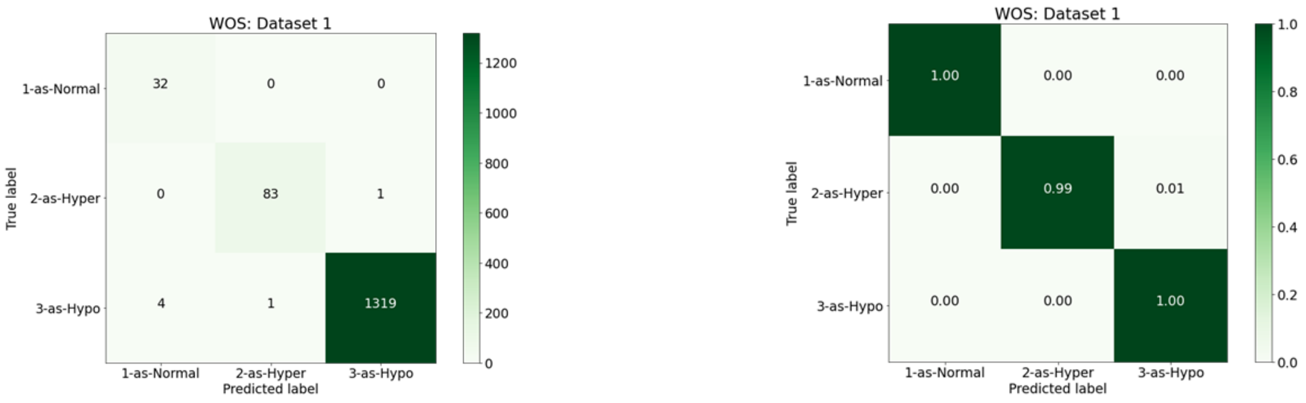
	Accuracy	**Precision**	**Recall**	**F1-score**
Original dataset (without ADASYN)	99.58 %	99.48 %	95.87 %	97.56%
Resampled dataset (with ADASYN resampling)	99.90 %	99.89 %	99.90 %	99.90%

Table 4 shows the performance of the proposed methodology for the second dataset. The original dataset without ADASYN resampling had an accuracy of 99.602%, precision of 99.785%, recall of 97.413% and F1-score of 98.565%. After resampling without ADASYN, the accuracy improved to 99.714%, precision to 99.711%, recall to 99.717% and F1-score to 99.713%. This improvement in performance again demonstrates the effectiveness of the proposed methodology in dealing with imbalanced datasets.

Table 4. Performance of proposed methodology for second dataset.

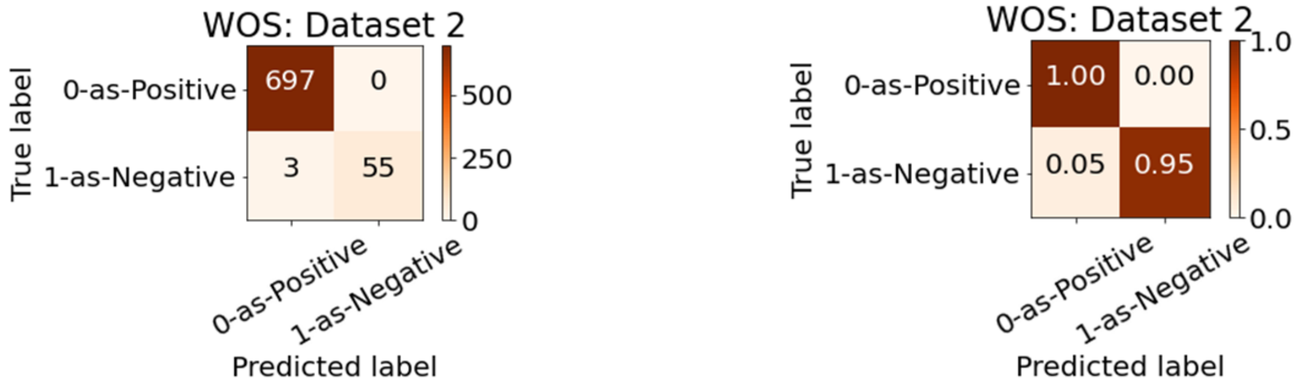
	Accuracy	Precision	Recall	F1-score
Original dataset (without ADASYN)	99.602 %	99.785 %	97.413 %	98.565 %
Resampled dataset (with ADASYN resampling)	99.714 %	99.711 %	99.717 %	99.713%

The confusion matrices for the first dataset before ADASYN resampling are illustrated in Figure 12. Similarly, Figure 13 illustrates the confusion matrices for the first dataset before ADASYN resampling.



(a) (b)

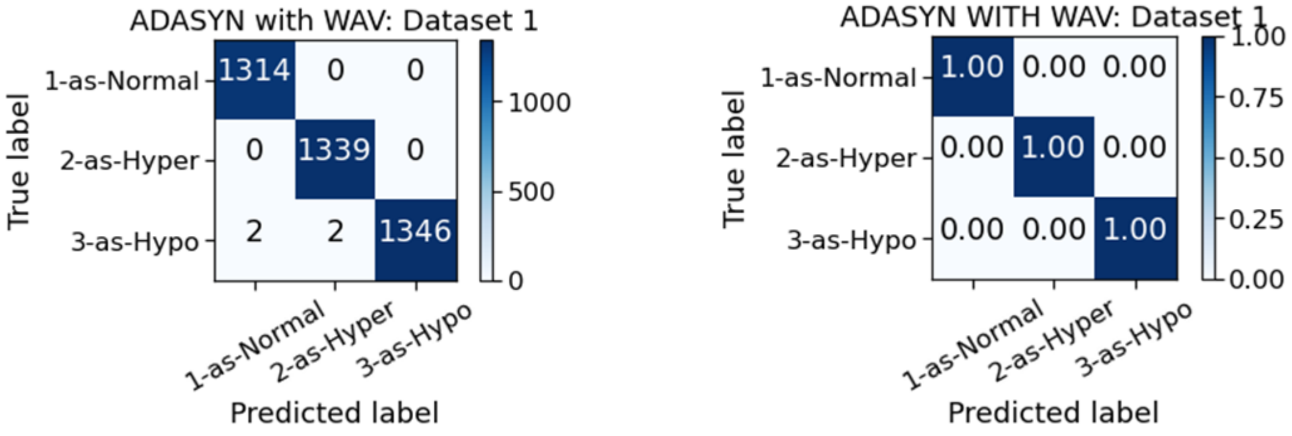
Figure 12. Confusion matrices of first dataset without resampling: (a) classification of number of test samples; (b) classification of test samples in percentage.



(a) (b)

Figure 13. Confusion matrices of second dataset without resampling: (a) classification of number of test samples; (b) classification of test samples in percentage

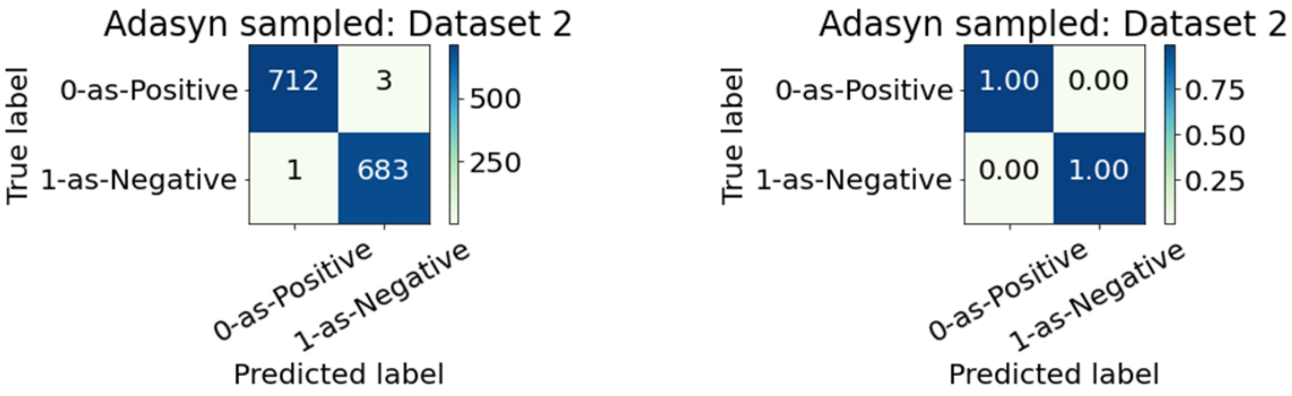
Figure 14 and Figure 15 show the confusion matrices for both datasets with ADASYN resampling and with weighted average voting. Figure 14 (a) shows the number of test samples classified as the model and Figure 14 (b) shows the results in percentage from first dataset. Similarly, Figure 15 (a) shows the number of test samples classified by the model and Figure 15 (b) shows the results in percentage from second dataset.



(a)

(b)

Figure 14. Confusion matrices of first dataset without resampling: (a) classification of number of test samples; (b) classification of test samples in percentage



(a)

(b)

Figure 15. Confusion matrices of first dataset without resampling: (a) classification of number of test samples; (b) classification of test samples in percentage

Comaprison with existing works

We have compared our proposed methodology with existing works done on the same dataset. Table 5 presents a comparison of our proposed methodology, which uses an ensemble, with existing studies on the first and second datasets. The performance metrics, such as accuracy, precision, recall and F1-score, are used to evaluate and compare the effectiveness of each technique employed in the respective studies.

For the first dataset (KEEL), our proposed SL ensemble method achieved an accuracy of 99.602%, precision of 99.785%, recall of 97.413% and F1-score of 98.565%. When compared to other studies on the same dataset, our methodology demonstrates superior performance in all metrics. For instance, study [54] using KNN achieved an accuracy of 98.600%, while study [55] employing KNN reached an accuracy of 96.900% and study [56] utilizing a liquid state machine (LSM) Autoencoder had an accuracy of 98.900%. Additionally, our method's precision and recall values outperform those reported in [56].

Table 5. Comparison of proposed methodology with existing studies on first dataset.

Ref	Year	Dataset	Technique	Accuracy	Precision	Recall	F1-score
This work	2023	KEEL	SL ensemble	99.602 %	99.785 %	97.413 %	98.565 %
[54]	2016	KEEL	KNN	98.600 %	-	-	-
[55]	2018	KEEL	KNN	96.900 %	-	-	-
[56]	2021	KEEL	LSM autoencoder	98.900 %	99.600 %	75.100 %	-

For the second dataset (hypothyroid), our proposed SL ensemble method achieved an accuracy of 99.714%, precision of 99.711%, recall of 99.717% and F1-score of 99.713%. When compared to other studies on the same dataset, our methodology again demonstrates superior performance in all metrics. Study [57] using KNN achieved an accuracy of 98.000%, while study [58] employing a random forest (RF) with sequential minimal optimization (SMO) reached an accuracy of 99.440% and study [59] utilizing a decision tree (DT) had an accuracy of 99.580%. It is worth noting that our method's recall value surpasses that reported in study [59]. As illustrated in Table 6, the proposed methodology, which employs an ensemble, demonstrates superior performance in terms of accuracy, precision, recall and F1-score when compared to other studies on both datasets.

Table 6. Comparison of proposed methodology with existing studies on second dataset.

Ref	Year	Dataset	Technique	Accuracy	Precision	Recall	F1-score
This work	2023	Hypothyroid	SL ensemble	99.714 %	99.711 %	99.717 %	99.713 %
[57]	2018	Hypothyroid	KNN	98.000 %	-	-	-
[58]	2021	Hypothyroid	RF with SMO	99.440 %	-	-	-
[59]	2022	Hypothyroid	DT	99.580 %	-	99.600 %	-

The proposed approach in the last also compared with the multiple distinct methodologies having a similar set of approach for the separate problem statement. For this purpose three studies have been selected . In [60] the Kernel slow feature analysis (KSFA) is implemented for the fault detection of the air unit. KSFA is a feature extraction approach that may capture time series data's temporal dynamics. KSFA can extract time-invariant slow features that may be utilized to enhance the performance of machine learning models using time series data. KSFA may be used to extract features from time series data in batches, which is beneficial when working with huge datasets. On the other hand, ADASYN can effectively produce additional training samples in order to build a somewhat balanced dataset and therefore get an efficient and robust prediction model. In [61] hybrid resampling technique (HRT) used with extreme learning machine ensemble. Both ADASYN and HRT are excellent oversampling approaches for enhancing machine learning model performance on unbalanced datasets. The approach used is determined by the unique use case and the type of the data. ADASYN is a computationally efficient and simple oversampling approach that can produce synthetic data for minority class instances adaptively. HRT is a hybrid resampling approach that uses oversampling and undersampling to balance the dataset and decrease model bias and variation. In comparison to ADASYN, HRT might be computationally costly. HRT may not be appropriate for all sorts of unbalanced datasets, but ADASYN is a computationally efficient and simple oversampling strategy. In [62] a technique of feature sparse representation is implemented. While feature sparse representation is intended to solve the issue of lack of features in machine learning, ADASYN is intended to address the issue of class imbalance. Understanding why feature sparse representation occurs is essential when constructing models since it may lead to issues like overfitting and less-than-ideal outcomes in learning models. The oversampling method ADASYN is useful for enhancing the performance of machine learning models on unbalanced datasets. A solution to the issue of sparse features in machine learning is feature sparse representation. The unique use case and the kind of data determine the approach to utilize.

5. Discussion

The research study presented in this paper aims to provide insights into the effectiveness of a proposed methodology for dealing with imbalanced datasets and the performance of an ensemble model in predicting thyroid disease outcomes. The results of the study demonstrate that addressing class imbalance through resampling techniques is an essential step in the preprocessing of imbalanced datasets, as it significantly improves the performance of machine learning models. The accuracy, precision, recall and F1-score showed significant improvements after applying ADASYN resampling in both datasets. This indicates that addressing class imbalance through resampling techniques is an essential step in the preprocessing of imbalanced datasets, as it improves the performance of the machine learning models. The results corroborate the importance of considering and addressing class imbalance in the data during the preprocessing stage.

The results demonstrate that the ensemble model, which combines multiple machine learning models, achieved higher prediction accuracy than any individual model alone. This finding supports the idea that combining the strengths of different models through ensemble techniques can lead to improved performance. The use of ensembles with distinct combinations of base and meta estimators further reinforces this notion, as it allows for leveraging the advantages of each model while mitigating their individual weaknesses. The weighted average voting technique, which assigns different weights to the models based on their performance, further improved the prediction accuracy of the ensemble model. This demonstrates that the incorporation of the weighted average voting technique helps to better capture the strengths of each model in the ensemble, leading to a more accurate and reliable prediction. The results obtained for both datasets indicate that the proposed methodology is not only effective in dealing with imbalanced datasets but also robust in predicting thyroid disease outcomes. The consistency in the improvement of performance metrics for both datasets demonstrate the potential of the methodology to be generalized and applied to other datasets with similar challenges.

For the first dataset (KEEL), our proposed SL ensemble method achieved an accuracy of 99.602%, precision of 99.785%, recall of 97.413% and F1-score of 98.565%. When compared to other studies on the same dataset, our methodology demonstrates superior performance in all metrics. The work

[54] employed KNN, a powerful and flexible class of models that have shown remarkable success in various tasks. However, their accuracy of 98.600% falls short compared to our SL ensemble. The work [55] used k-nearest neighbors (KNN), a simple yet effective algorithm for classification tasks, but only reached an accuracy of 96.900%. The work [56] utilized a LSM autoencoder, a bio-inspired neural network model and achieved an accuracy of 98.900%. The precision and recall values reported in the paper [56] are lower than those in our method, which indicates that our method has better discriminatory power between the classes.

For the second dataset (hypothyroid), our proposed SL ensemble method achieved an accuracy of 99.714%, precision of 99.711%, recall of 99.717% and F1-score of 99.713%. When compared to other studies on the same dataset, our methodology again demonstrates superior performance in all metrics. The work [57] used KNN and achieved an accuracy of 98.000%, which is lower than our SL ensemble. The paper [58] employed a random forest (RF) with sequential minimal optimization (SMO), a combination of a powerful ensemble method and a technique for solving large-scale optimization problems. However, their accuracy of 99.440% is still lower than ours. Study [59] utilized a decision tree (DT), a popular and interpretable machine learning model and achieved an accuracy of 99.580%. Although their recall value is comparable to our method, our method still outperforms the study [59] in accuracy, precision and F1-score.

However, it is important to consider some limitations of the research study and potential areas for improvement. While the ensemble model showed improved performance, it may be computationally expensive due to the use of multiple base estimators and iterations of the super learner technique. Future research could explore methods to optimize the computational efficiency of the ensemble model without compromising its performance. Additionally, the selection of base and meta estimators in the super learner ensembles was based on their individual strengths and potential synergies when combined. However, the optimal combination of models may vary depending on the dataset and the problem at hand.

6. Conclusions

The proposed methodology for thyroid cancer classification using a super learner ensemble model with resampling techniques and weighted average voting showed significant improvements in performance on imbalanced datasets. The results demonstrate the importance of addressing class imbalance in the data during the preprocessing stage and the benefits of combining multiple machine learning models for improving prediction accuracy. The super learner ensemble method achieved higher prediction accuracy than any individual model alone and the use of distinct combinations of base and meta estimators further improved performance. The proposed methodology showed superior performance compared to other studies on the same datasets, demonstrating its potential to be applied to other datasets with similar challenges. However, the computational complexity of the ensemble model and the optimal selection of base and meta estimators remain as limitations that require further research. Overall, the proposed methodology shows promise in improving the accuracy of thyroid cancer classification and can potentially aid in the diagnosis and treatment of thyroid cancer patients.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was funded by Princess Nourah bint Abdulrahman University and Researchers Supporting Project number (PNURSP2023R346), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflict of interest

The authors declare no conflict of interest.

References

1. S. Grodski, T. Brown, S. Sidhu, A. Gill, B. Robinson, D. Learoyd, et al., Increasing incidence of thyroid cancer is due to increased pathologic detection, *Surgery*, **144** (2008), 1038–1043.
2. J. Kim, J. E. Gosnell, S. A. Roman, Geographic influences in the global rise of thyroid cancer, *Nat. Rev. Endocrinol.*, **16** (2020), 17–29.
3. H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries, *CA: Cancer J. Clin.*, **71** (2021), 209–249. https://doi.org/10.3322/caac.21660
4. L. Enewold, K. Zhu, E. Ron, A. J. Marrogi, A. Stojadinovic, G. E. Peoples, et al., Rising thyroid cancer incidence in the United States by demographic and tumor characteristics, 1980–2005, *Cancer Epidem. Biomar.*, **18** (2009), 784–791. https://doi.org/10.1109/JMEMS.2009.2023841
5. L. Davies, H. G. Welch, Current thyroid cancer trends in the United States, *JAMA Otolaryngology-Head Neck Surgery*, **140** (2014), 317. https://doi.org/10.1016/j.neucom.2014.03.007
6. P. B. Manoj, A. Innisai, D. S. Hameed, A. Khader, M. Gopanraj, N. H. Ihare, Correlation of high-resolution ultrasonography findings of thyroid nodules with ultrasound-guided fine-needle aspiration cytology in detecting malignant nodules: A retrospective study in Malabar region of Kerala, South India, *J. Fam. Med. Prim. Care*, **8** (2019), 1613.
7. H. Tan, Z. Li, N. Li, J. Qian, F. Fan, H. Zhong, et al., Thyroid imaging reporting and data system combined with Bethesda classification in qualitative thyroid nodule diagnosis, *Medicine*, **98** (2019), 2019.
8. A. N. Rajalakshmi, F. Begam, Thyroid Hormones in the Human Body: A review, *J. Drug Delivery Ther.*, **11** (2021), 178–182. https://doi.org/10.22270/jddt.v11i5.5039
9. A. K. Lee, P. M. A. Tacanay, P. Siy, D. T. Argamosa, Ectopic papillary thyroid carcinoma presenting as right lateral neck mass, *JAFES*, **37** (2022), 2022.
10. M. I. Larg, D. Apostu, C. Petean, K. Gabora, I. C. Bdulescu, E. Olariu, et al., Evaluation of malignancy risk in 18F-FDG PET/CT thyroid incidentalomas, *Diagnostics*, **9** (2019), 92. https://doi.org/10.3390/diagnostics9030092

11. M. Hanan, E. Fatma, A. Aly, A. Medhat, Evaluation of Incidental Thyroid Findings Detected by Positron Emission Tomography/Computed Tomography, *Medical J. Cairo University*, **87** (2019), 819–826. https://doi.org/10.21608/mjcu.2019.52541

12. S. Quazi, Artificial intelligence and machine learning in precision and genomic medicine, *Med. Oncol.*, **39** (2022), 120.

13. K. Preuss, N. Thach, X. Liang, M. Baine, J. Chen, C. Zhang, et al., Using quantitative imaging for personalized medicine in pancreatic cancer: a review of radiomics and deep learning applications, *Cancers*, **14** (2022), 1654. https://doi.org/10.3390/cancers14071654

14. N. Shusharina, D. Yukhnenko, S. Botman, V. Sapunov, V. Savinov, G. Kamyshov, et al., Modern methods of diagnostics and treatment of neurodegenerative diseases and depression, *Diagnostics*, **13** (2023), 573. https://doi.org/10.3390/diagnostics13030573

15. S. Khalil, U. Nawaz, Zubariah, Z. Mushtaq, S. Arif, M. Z. ur Rehman, et al., Enhancing ductal carcinoma Classification using transfer learning with 3D U-Net models in breast cancer imaging, *Appl. Sci.*, **13** (2023), 4255.

16. A. M. Antoniadi, Y. Du, Y. Guendouz, L. Wei, C. Mazo, B. A. Becker, et al., Current challenges and future opportunities for XAI in machine learning-based clinical decision support systems: A systematic review, *Appl. Sci.*, **11** (2021), 5088.

17. Z. Mushtaq, M. F. Qureshi, M. J. Abbass, S. M. Q. AlFakih, Effective kernelprincipal component analysis based approach for wisconsin breast cancer diagnosis, *Electron. Lett.*, **59** (2023).

18. X. M. Keutgen, H. Li, K. Memeh, J. Conn Busch, J. Williams, L. Lan, D. Sarne, et al., A machine-learning algorithm for distinguishing malignant from benign indeterminate thyroid nodules using ultrasound radiomic features, *J. Med. Imaging*, **9** (2022), 034501–034501.

19. V. V. Vadhiraj, A. Simpkin, J. OConnell, N. Singh Ospina, S. Maraka, D. T. OKeeffe, Ultrasound image classification of thyroid nodules using machine learning techniques, *Medicina*, **57** (2021), 527. https://doi.org/10.3390/medicina57060527

20. M. Bereby-Kahane, R. Dautry, E. Matzner-Lober, F. Cornelis, D. Sebbag-Sfez, V. Place, et al., Prediction of tumor grade and lymphovascular space invasion in endometrial adenocarcinoma with MR imaging-based radiomic analysis, *Diagn. Interv. Imag.*, **101** (2020), 401–411.

21. K. E. Fasmer, E. Hodneland, J. A. Dybvik, K. Wagner-Larsen, J. Trovik, A. Salvesen, et al., Whole-volume tumor MRI radiomics for prognostic modeling in endometrial cancer, *J. Magn. Reson. Imaging*, **53** (2021), 928–937.

22. A. Prete, P. Borges de Souza, S. Censi, M. Muzza, N. Nucci, M. Sponziello, Update on fundamental mechanisms of thyroid cancer, *Front. Endocrinol.*, **11** (2020), 102.

23. N. Pozdeyev, M. M. Rose, D. W. Bowles, R. E. Schweppe, Molecular therapeutics for anaplastic thyroid cancer, In: *Seminars in Cancer Biology*, **61** (2020), 23–29. https://doi.org/10.1016/j.semcancer.2020.01.005

24. Y. C. Zhu, P. F. Jin, J. Bao, Q. Jiang, X. Wang, Thyroid ultrasound image classification using a convolutional neural network, *Ann. Transl. Med.*, **9** (2021).

25. M. R. Kwon, J. H. Shin, H. Park, H. Cho, S. Y. Hahn, K. W. Park, Radiomics study of thyroid ultrasound for predicting BRAF mutation in papillary thyroid carcinoma: Preliminary results, *Am. J. Neuroradiol.*, **41** (2020), 700–705. https://doi.org/10.3174/ajnr.A6505

26. Y. Wang, W. Yue, X. Li, S. Liu, L. Guo, H. Xu, et al., Comparison study of radiomics and deep learning-based methods for thyroid nodules classification using ultrasound images, *Ieee Access*, **8** (2020), 52010–52017.

27. D. Chen, J. Hu, M. Zhu, N. Tang, Y. Yang, Y. Feng, Diagnosis of thyroid nodules for ultrasonographic characteristics indicative of malignancy using random forest, *BioData Min.*, **13** (2020), 1–21.

28. H. K. Shivastuti, J. Manhas, V. Sharma, Performance evaluation of SVM and random forest for the diagnosis of thyroid disorder, *Int. J. Res. Appl. Sci. Eng. Technol.*, **9** (2021), 945–947.

29. H. Abbad Ur Rehman, C. Y. Lin, Z. Mushtaq, Effective K-nearest neighbor algorithms performance analysis of thyroid disease, *J. Chin. Inst. Eng.*, **44** (2021), 77–87. https://doi.org/10.14358/PERS.87.2.77

30. T. Akhtar, S. O. Gilani, Z. Mushtaq, S. Arif, M. Jamil, Y. Ayaz, et al., Effective voting ensemble of homogenous ensembling with multiple attribute-selection approaches for improved identification of thyroid disorder, *Electronics*, **10** (2021), 3026.

31. L. C. Zhu, Y. L. Ye, W. H. Luo, M. Su, H. P. Wei, X. B. Zhang, et al., A model to discriminate malignant from benign thyroid nodules using artificial neural network, *PLoS One*, **8** (2013), e82211. https://doi.org/10.1371/journal.pone.0082211

32. B. Zhang, J. Tian, S. Pei, Y. Chen, X. He, Y. Dong, et al., Machine learningassisted system for thyroid nodule diagnosis, *Thyroid*, **29** (2019), 858–867. https://doi.org/10.1089/thy.2018.0380

33. A. K. Singh, A comparative study on disease classification using machine learning algorithms, In *Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE)*, 2019.

34. E. Sonu, Thyroid disease classification using machine learning algorithms, In: *Journal of Physics: Conference Series*, vol. 1963, p. 012140, IOP Publishing, 2021. https://doi.org/10.1088/1742-6596/1963/1/012140

35. P. Poudel, A. Illanes, E. J. Ataide, N. Esmaeili, S. Balakrishnan, M. Friebe, Thyroid ultrasound texture classification using autoregressive features in conjunction with machine learning approaches, *IEEE Access*, **7** (2019), 79354–79365. https://doi.org/10.1109/ACCESS.2019.2923547

36. D. C. Yadav, S. Pal, Thyroid prediction using ensemble data mining techniques, *Int. J. Inf. Technol.*, **14** (2022), 1273–1283.

37. S. S. Z. Mousavi, M. M. Zanjireh, M. Oghbaie, Applying computational classification methods to diagnose Congenital Hypothyroidism: A comparative study, *Inf. Medicine Unlocked*, **18** (2020), 100281.

38. D. T. Nguyen, J. K. Kang, T. D. Pham, G. Batchuluun, K. R. Park, Ultrasound image-based diagnosis of malignant thyroid nodule using artificial intelligence, *Sensors*, **20** (2020), 1822. https://doi.org/10.3390/s20071822

39. G. Chaubey, D. Bisen, S. Arjaria, V. Yadav, Thyroid disease prediction using machine learning approaches, *Natl. Acad. Sci. Lett.*, **44** (2021), 233–238.

40. M. Garcia de Lomana, A. G. Weber, B. Birk, R. Landsiedel, J. Achenbach, K. J. Schleifer, et al., In silico models to predict the perturbation of molecular initiating events related to thyroid hormone homeostasis, *Chem. Res. Toxicol.*, **34** (2020), 396–411.

41. K. Shankar, S. K. Lakshmanaprabu, D. Gupta, A. Maseleno, V. H. C. De Albuquerque, Optimal feature-based multi-kernel SVM approach for thyroid disease classification, *J. Supercomput.*, **76** (2020), 1128–1143.

42. H. Abbad Ur Rehman, C. Y. Lin, Z. Mushtaq, S. F. Su, Performance analysis of machine learning algorithms for thyroid disease, *Arab. J. Sci. Eng.*, 1–13, 2021.

43. R. Das, S. Saraswat, D. Chandel, S. Karan, J. S. Kirar, An AI Driven Approach for Multiclass Hypothyroidism Classification, In: *Advanced Network Technologies and Intelligent Computing: First International Conference, ANTIC 2021, Varanasi, India, December 1718, 2021, Proceedings*, pp. 319–327, Springer, 2022.

44. M. Hosseinzadeh, O. H. Ahmed, M. Y. Ghafour, F. Safara, H. K. Hama, S. Ali, et al., A multiple multilayer perceptron neural network with an adaptive learning algorithm for thyroid disease diagnosis in the internet of medical things, *J. Supercomput.*, **77** (2021), 3616–3637.

45. M. Riajuliislam, K. Z. Rahim, A. Mahmud, Prediction of Thyroid Disease (Hypothyroid) in Early Stage Using Feature Selection and Classification Techniques, In: *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, pp. 60–64, IEEE, 2021.

46. R. Jha, V. Bhattacharjee, A. Mustafi, Increasing the prediction accuracy for thyroid disease: A step towards better health for society, *Wireless Pers. Commun.*, **122** (2022), 1921–1938. https://doi.org/10.1155/2022/9809932

47. T. Alyas, M. Hamid, K. Alissa, T. Faiz, N. Tabassum, A. Ahmad, Empirical method for thyroid disease classification using a machine learning approach, *BioMed Res. Int.*, **22** (2022).

48. S. Sankar, A. Potti, G. N. Chandrika, S. Ramasubbareddy, Thyroid disease prediction using XGBoost algorithms, *J. Mob. Multimed*, **18** (2022), 1–18.

49. I. Ali, Z. Mushtaq, S. Arif, A. Algarni, N. Soliman, W. El-Shafai, Hyperspectral images-based crop classification scheme for agricultural remote sensing, *Comput. Syst. Sci. Eng.*, **46** (2023), 303–319.

50. S. Arif, S. Munawar, H. Ali, Driving drowsiness detection using spectral signatures of EEG-based neurophysiology, *Front. Physiol.*, **14** (2023), 1153268.

51. S. Arif, M. Arif, S. Munawar, Y. Ayaz, M. J. Khan, N. Naseer, EEG spectral comparison between occipital and prefrontal cortices for early detection of driver drowsiness, In: *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, pp. 1–6, IEEE, 2021.

52. S. Arif, M. J. Khan, N. Naseer, K. S. Hong, H. Sajid, Y. Ayaz, Vector phase analysis approach for sleep stage classification: A functional near-infrared spectroscopy-based passive braincomputer interface, *Front. Hum. Neurosci.*, **15** (2021), 658444.

53. T. Akhtar, S. Arif, Z. Mushtaq, S. O. Gilani, M. Jamil, Y. Ayaz, et al., Ensemble-based effective diagnosis of thyroid disorder with various feature selection techniques, In: *2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, pp. 14–19, IEEE, 2022.

54. K. Chandel, V. Kunwar, S. Sabitha, T. Choudhury, S. Mukherjee, A comparative study on thyroid disease detection using K-nearest neighbor and Naive Bayes classification techniques, *CSI Transactions ICT*, **4** (2016), 313–319. https://doi.org/10.1111/twec.13285

55. R. Pal, T. Anand, S. K. Dubey, Evaluation and performance analysis of classification techniques for thyroid detection, *Int. J. Bus. Inf. Syst.*, **28** (2018), 163–177.

56. M. Saktheeswari, T. Balasubramanian, Multi-layer tree liquid state machine recurrent auto encoder for thyroid detection, *Multimed. Tools Appl.*, **80** (2021), 17773–17783. https://doi.org/10.1007/s11042-020-10243-7

57. A. Tyagi, R. Mehra, A. Saxena, Interactive Thyroid Disease Prediction System Using Machine Learning Technique, In: *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, (Solan Himachal Pradesh, India), pp. 689–693, IEEE, Dec. 2018.

58. S. Mishra, Y. Tadesse, A. Dash, L. Jena, P. Ranjan, Thyroid Disorder Analysis Using Random Forest Classifier, In: *Intelligent and Cloud Computing* (D. Mishra, R. Buyya, P. Mohapatra, and S. Patnaik, eds.), Smart Innovation, Systems and Technologies, (Singapore), pp. 385–390, Springer, 2021.

59. K. Guleria, S. Sharma, S. Kumar, S. Tiwari, Early prediction of hypothyroidism and multiclass classification using predictive machine learning and deep learning, *Measurement: Sensors*, **24** (2022), 100482. https://doi.org/10.1016/j.measen.2022.100482

60. H. Zhang, C. Li, D. Li, Y. Zhang, W. Peng, Fault detection and diagnosis of the air handling unit via an enhanced kernel slow feature analysis approach considering the time-wise and batch-wise dynamics, *Energ. Buildings*, **253** (2021), 111467. https://doi.org/10.1016/j.enbuild.2021.111467

61. H. Zhang, W. Yang, W. Yi, J. B. Lim, Z. An, C. Li, Imbalanced data based fault diagnosis of the chiller via integrating a new resampling technique with an improved ensemble extreme learning machine, *J. Build. Eng.*, **70** (2023), 106338. https://doi.org/10.1016/j.jobe.2023.106338

62. H. Zhang, C. Li, Q. Wei, Y. Zhang, Fault detection and diagnosis of the air handling unit via combining the feature sparse representation based dynamic SFA and the LSTM network, *Energ. Buildings*, **269** (2022), 112241.