

RESEARCH

Open Access



# Machine learning framework with feature selection approaches for thyroid disease classification and associated risk factors identification

Azrin Sultana<sup>1</sup> and Rakibul Islam<sup>1\*</sup>

\*Correspondence:  
rakibulislam.cse21@gmail.com

<sup>1</sup> Department of Computer  
Science, American International  
University-Bangladesh,  
Dhaka 1229, Bangladesh

## Abstract

Thyroid disease (TD) develops when the thyroid does not generate an adequate quantity of thyroid hormones as well as when a lump or nodule emerges due to aberrant growth of the thyroid gland. As a result, early detection was pertinent in preventing or minimizing the impact of this disease. In this study, different machine learning (ML) algorithms with a combination of scaling method, oversampling technique, and various feature selection approaches have been applied to make an efficient framework to classify TD. In addition, significant risk factors of TD were also identified in this proposed system. The dataset was collected from the University of California Irvine (UCI) repository for this research. After that, in the preprocessing stage, Synthetic Minority Oversampling Technique (SMOTE) was used to resolve the imbalance class problem and robust scaling technique was used to scale the dataset. The Boruta, Recursive Feature Elimination (RFE), and Least Absolute Shrinkage and Selection Operator (LASSO) approaches were used to select appropriate features. To train the model, we employed six different ML classifiers: Support Vector Machine (SVM), AdaBoost (AB), Decision Tree (DT), Gradient Boosting (GB), K-Nearest Neighbors (KNN), and Random Forest (RF). The models were examined using a 5-fold CV. Different performance metrics were observed to compare the effectiveness of the algorithms. The system achieved the most accurate results using the RF classifier, with 99% accuracy. This proposed system will be beneficial for physicians and patients to classify TD as well as to learn about the associated risk factors of TD.

**Keywords:** Thyroid disease prediction, Random forest, Healthcare, Machine learning, Feature selection

## Introduction

TD alludes to a condition in which the thyroid, a teensy, butterfly-shaped hormone-producing gland, emits either an excessive amount or an insufficient amount of these pertinent hormones [1, 2]. TD is categorized into four types: hyperthyroidism, hypothyroidism, thyroiditis, and Hashimoto's thyroiditis [3–5]. A significant portion of adult women (between 9 and 15 percent) and men (a lesser percentage)

are thought to be affected. Numerous thyroid complications affect more than 20 million Americans, according to one study [6]. Experts estimate that around 12% of individuals will suffer from a thyroid disorder at some point in their lifetime. Women are approximately five to eight times more likely than males to have thyroid disorders, and one woman out of every eight will experience a thyroid condition over her lifespan [7]. Both the size and effectiveness of the thyroid gland fluctuate substantially throughout pregnancy. During pregnancy, the gland expands 10% broader in iodine-rich regions and 20–40% wider in iodine-deficient areas [8]. Hormones that regulate how the body utilizes energy are produced primarily from ingested iodine [9]. However, about a third of the entire population is concentrated in places with insufficient iodine [10, 11].

Thyroid hormone is responsible for the production and secretion of triiodothyronine (T3) and thyroxine (T4), the only iodine-containing hormones in vertebrates [12–14]. These hormones are mandated for adequate growth, differentiation, and metabolic regulation [15]. The anterior pituitary gland is responsible for synthesizing serum thyrotrophin (TSH), which is responsible for regulating the production of these hormones [16]. Approximately 95% of the thyroid hormone present in the blood is T4, which regulates metabolism, temperament, and the body's core temperature. Conventionally, T3 compensates for 5% of the thyroid hormone discovered in the blood [17, 18]. Hyperthyroidism, or perhaps an overactive thyroid, may cause a wide variety of adverse consequences. Some of these include vision loss, irregular heartbeat, fragile bones, and skin that becomes easily irritated or inflamed. Hypothyroidism, or an underactive thyroid, can lead to a wide range of side effects, such as an oversized thyroid, or goiter, which can impair breathing and ingesting, high blood cholesterol, cardiovascular disease, damaged nerves that cause itching, fertility problems, congenital abnormalities, and anxiety [19, 20]. By making predictions at an early stage, we may be able to change the course of TD, alleviate symptoms, and avert the irreversible effects. The severity of TD lessens grievous complications, and enhanced patient safety will result from an accurate assessment by utilizing ML algorithms [21].

Arthur Samuel defines ML as the discipline that facilitates computers to acquire knowledge without even being explicitly programmed [22]. In order to draw insights from historical data and identify meaningful patterns within complex, unorganized datasets, ML algorithms use a wide collection of statistical, probabilistic, and optimization techniques [23–25]. Automatically classifying text [26, 27], finding network intrusions [28, 29], figuring out what customers buy [29], predicting diseases [30], and giving a decision support system [31] are just some of the many uses for these algorithms. ML makes predictions within an acceptable range by using preprogrammed algorithms that acquire knowledge from their input data and evaluate it to enhance their performance.

It is skeptical about choosing important attributes that may be employed as risk factors in prediction frameworks. To construct a reliable predictive model, it is important to carefully select the most suitable optimal combination of parameters and ML algorithms [32]. To determine which features are most important, this research aims to make use of the thyroid dataset and three different feature selection methods: RFE, Boruta, and

LASSO. This aids in eliminating the ML issues of overfitting and underfitting. In this study, we used a wide range of supervised models, including AB, DT, GB, KNN, and RF for TD classification. The key contributions of this study are as follows:

- With the use of ML algorithms, we devised a reliable approach for assessing whether a particular patient is suffering from TD.
- Exploring the most prominent contributing factors of TD.
- Several feature selection approaches, including RFE, Boruta, and LASSO, have been employed to extract the most pertinent features from the dataset, impacting the ML algorithms' performance.
- The performance metrics of different models are also evaluated in this study.

### **Related work**

Using data mining meta-classification methods which include boosting, bagging, stacking, and voting with a novel ensemble classifier, the authors of [33] evaluated TD on an extensive and convoluted dataset while comparing accuracy, sensitivity, and specificity. The authors conducted their suggested approach through two rounds of experiments to determine which of them produced the best outcomes. They compared the system's performance metrics using several k values, such as 10, 15, and 20. They also explored the dataset using a variety of splits between the train and test sets. Thus, the use of different k-values in training and testing data helped them to improve the efficiency of the algorithms employed in this investigation.

Predictive treatment for TD is the subject of the study reported in [34]. ML methods are used to determine, depending on thyroid hormone parameters and other clinical information about the patient, whether the patient's therapy ought to be increased, lowered, or left unchanged. This research aims to forecast the synthetic thyroid hormone therapy direction for hypothyroid individuals. Using SMOTE for pre-processing the dataset and the extra tree classifier (ETC) as an ML model, the best outcomes are achieved by partitioning the data and balancing it. The parameters produced an F1-Score of 84%, a Precision of 84%, a Recall of 84%, and an Accuracy of 84%.

Authors in [35] use multiple ML methods on the dataset to construct a comparison study to better classify TD depending on dataset attributes. In order to provide precise predictions for the categorization, the dataset has also been altered. The classification was run on both the sampled and undersampled samples to provide more precise and reliable comparisons. Finally, the authors achieved 94.8% accuracy on the RF method, the maximum accuracy conceivable with this process.

H.A.U. Rehman et al. [36] analyze accuracy and other performance assessment criteria to predict and diagnose TD employing five distinct ML algorithms: KNN, DT, SVM, logistic regression (LR), and Naive Bayes (NB). To achieve optimal accuracy and performance in the initial phase of the experiment, feature selection approaches are omitted. The second and third stages of the research introduced feature selection strategies based on L1 and L2, respectively. When compared to results obtained

both with and without feature selection, L1-based feature selection yields the highest accuracy.

Using the UCI thyroid dataset, M. D. Maysanjaya et al. [37] employed six distinct types of ML algorithms to predict TD and evaluated the degree of accuracy among numerous artificial neural network approaches for classifying the type of thyroid gland into three categories, respectively, normal, hyperthyroid, and hypothyroid. The 10-fold cross-validation (CV) approach was employed in this investigation. The multilayer perceptron (MLP) approach outperforms others in terms of accuracy, recall, and the F1 measure.

Ahmed et al. [38] introduced a comprehensive intelligent hybrid model for the identification of TD utilizing linear discriminant analysis (LDA), KNN weighted preprocessing, and an adaptive neurofuzzy inference system (ANFIS). The entire model is comprised of three distinct phases. The LDA portion of the LDA-KNN-ANFIS model initially uses dimensionality reduction to get rid of extraneous characteristics in the disease dataset. Phase two involves applying a KNN-based weighted preprocessor on the input characteristics. In the last phase, preprocessed attributes are supplied to the ANFIS system for prediction.

An ML-based TD prediction framework focusing on the multi-class problem is presented in [39]. The research explores using a feature engineering strategy combined with an ETC model. Based on their observed effectiveness for disease prediction, five ML algorithms are evaluated; moreover, three DL methods with 16-bath and 100-epochs are also implemented in this study. In terms of the confusion matrix, 10-fold CV, standard deviation, accuracy level, precision, recall rates, and F1 score, several performance assessment approaches are assessed.

Overall, the literature review exhibits that many individuals have contributed to the TD prediction model. However, the main research gap we discover is that the majority of researchers exclusively work only with predictive models. Most of the previous researchers did not use the proper scaling method and also did not resolve the imbalanced class problem of this dataset. To address these constraints, this work proposes a strategy for balancing the dataset by using SMOTE and analyzes the optimal subset of features by using several feature selection techniques to apply ML approaches, offering a highly accurate TD classification solution and providing a complete comparison of the performance of ML-based systems that also compel an aspect to expand the understanding of related risk factors of TD. Various future directions have also been addressed.

## Methodology

### Dataset description

The dataset is retrieved from the UCI ML Repository [40]. It has 2,800 instances and 28 characteristics. Twenty of the 28 characteristics are categorical; these include query on thyroxine, on antithyroid medication, sick, pregnant, thyroid surgery, I131 treatment, query hypothyroid, query hyperthyroid, lithium, goiter, tumor, hypopituitary, psych, TSH measured, T3 measured, TT4 measured, T4U measured, FTI measured, TBG measured, and referral are categorical. Most of them

**Table 1** Continuous values of the dataset

Serial No	Attributes name	Attribute types	Range or type
1	age	Age of the patients	1–455
2	TSH	Thyroid-stimulating hormone	0.1–39
3	T3	Triiodothyronine	0.05–10
4	TT4	free thyroxine hormone level in the blood	2–430
5	T4U	thyroxine utilization rate	0.31–2.12
6	FTI	free T4 index for diagnosing TD	2–345

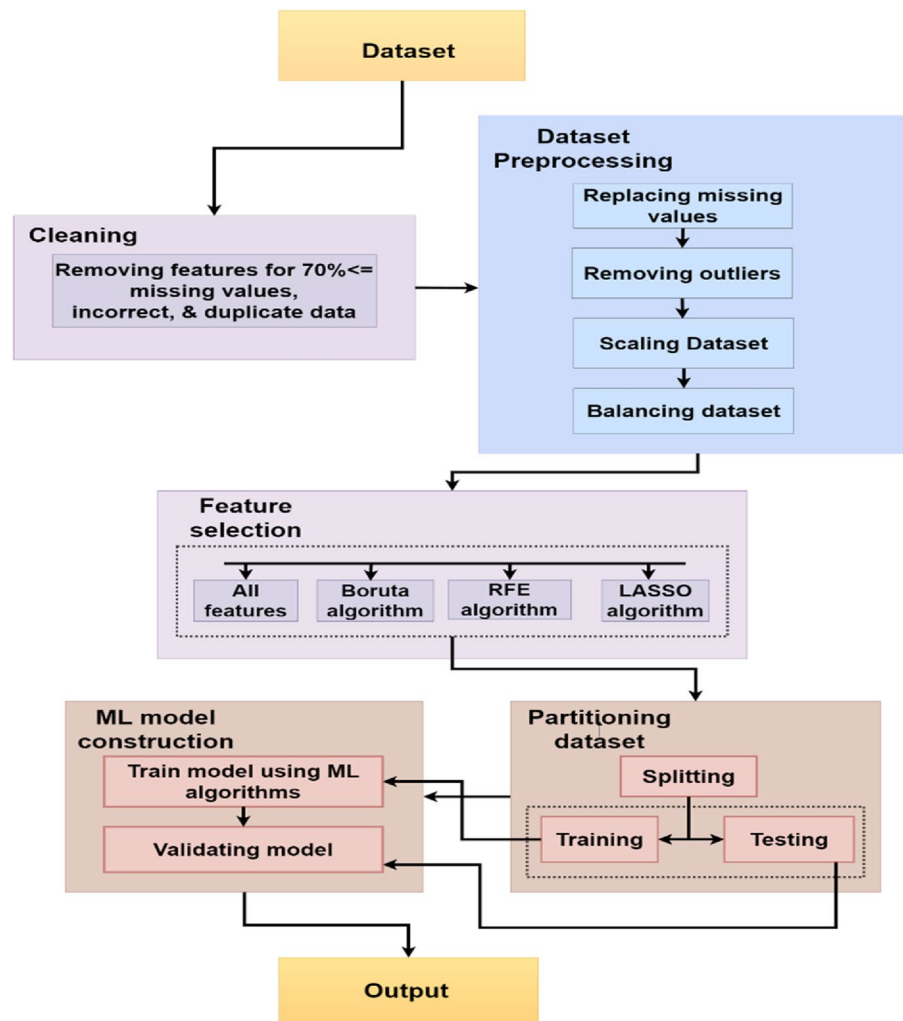
are categorical which are denoted by true and false. In addition, six attributes are continuous and are illustrated in Table 1. Based on the diagnostic findings, the total patient population is split into two groups: hyperthyroidism, represented by one, and normal, represented by 0. TD is present in 77 of the 2800 samples, while 2723 are thyroid-negative. Some features have missing values, which are defined by the question mark ("?").

#### Model diagram

In this study, not only a highly predictive model but also the key characteristics of TD were identified. All the experiments were conducted on a laptop with 7<sup>th</sup> generation Intel Core i5 processor with a 8 GB RAM. All the necessary code was written in Python v3.9.10 and implemented in Jupiter Notebook. Dataset analysis was performed with the help of the Sklearn, Matplotlib, Pandas, and Numpy library. There are seven steps involved in this system's workflow: data acquisition, data cleansing, dataset preprocessing, feature engineering, dataset splitting, model development, and outcome prediction. The main procedure of the framework began with data collecting. During the data cleaning phase, duplicate rows and features with more than 70% missing values were discarded, including any rows with just complete null or zero values. The dataset was scaled after missing values were imputed with the median value and outliers were eliminated; after that, the dataset was balanced in the final stage of preprocessing. The feature engineering stage, the fifth step of our system, consists of three feature selection methods for choosing the best features. Following that, the dataset was partitioned using an 80:20 ratio. The model was trained and tested using a variety of ML approaches in the sixth step of our proposed system. Finally, the classifier determines whether or not a person has TD. Figure 1 depicts the proposed system's model diagram.

#### Preprocessing

ML algorithm's ability to generalize performance is always significantly impacted by the data preparation. Datasets utilized in research tend to be flawed due to the presence of missing values, noise, and distortions [41]. Due to the inconsistent data, the dataset is skewed, and the ML algorithms find it challenging to make accurate predictions [42]. The dataset used to conduct this study also comprises some missing values and atypical values. The preparation of the dataset involved using a variety of



**Fig. 1** Model diagram of the proposed system

approaches. We eliminated the "TBG" column from the dataset since it contained over 70% missing values and is not providing accurate information. The median values are used to replace the missing data, and 0 and 1 are used to represent the two categories in this study. Table 1 exemplifies the presence of outliers in this dataset. As a result, it is preferable to discover and eliminate outliers. The interquartile range (IQR) is used to eliminate outliers from the dataset. Furthermore, when ML methods are relied on Euclidean distance [43, 44], feature scaling is a crucial component of the preprocessing. The robust scaler approach is used to scale attributes after missing data, and outliers are removed [45]. SMOTE employs to balance the dataset since it was highly unbalanced.

### **Feature selection**

Overfitting, learning accuracy, calculation time, and improved model learning may all be addressed with feature selection. Feature selection usually is the process of selecting the most important features from a dataset while eliminating redundant or unnecessary ones in order to improve classification accuracy, reduced processing cost, and extract the best features for classification [46, 47]. Boruta, RFE, and LASSO are the three feature selection approaches employed in our proposed system to determine the optimal subset of features for optimized performance.

#### ***Boruta algorithm***

A wrapper that was developed around the RF classification technique is known as the Boruta algorithm. It employs the Z score as the indicator of importance. However, the Z score cannot be utilized to determine the significance of any particular character because it requires some external reference. To do this, random properties need to be added to the information system. Have a "shadow" property that corresponds to each randomly chosen attribute and gets its value by redistributing the importance of the initial attribute across instances [48]. Train the system with an RF classifier after adding these shadow characteristics to the original dataset. The most relevant original characteristics of the model are any that are more significant than the most notable shadow feature. [49–51].

#### ***Recursive feature elimination (RFE)***

A common method for choosing pertinent characteristics is the RFE technique. It is a strategy for simplifying a model by selecting its most salient attributes and rejecting those that are less relevant [52]. The selection process narrows down the list of attributes by gradually removing features that are not important for achieving optimal performance. The estimating model is trained using the original set of features, and the importance of each result is then determined by applying some arbitrary attribute or callable. Following that, the least important features are purged from the current collection of attributes. After that, the procedure is repeated recursively on the condensed collection until the desired number of features to be chosen is accomplished. The best-scoring feature set is chosen by combining CV with RFE to score several feature subsets and discern the ideal features [53, 54].

#### ***Least absolute shrinkage and selection operator (LASSO)***

LASSO enables effective feature selection utilizing the linear relationship between input attributes and target output [55]. The coefficients may be quickly shrunk and removed to decrease variance, allowing for highly precise predictions. To achieve this goal of minimization, LASSO regression will selectively retain just the most informative attributes while dropping the rest [56, 57]. This operator's capability to perform minimal selection and shrinking depends on changing the absolute amount of the coefficient among functions. It is possible to eliminate features from consideration if their coefficient value is zero, and attributes with negative coefficients



are also able to be omitted. Cost functions have a positive relationship with the coefficient of a feature. Therefore, the goal of LASSO regression is to reduce the absolute values of the coefficients while still optimizing the cost function. After the shrinking procedure is complete, the variables with the largest remaining nonzero coefficients are chosen as model features [58, 59]. The cost function of LASSO feature selection is as follows:

$$a(\theta) = \frac{1}{R} \sum_{i=1}^m (y_i - y_k)^2 + \alpha \sum_{k=1}^R |a_k| \quad (1)$$

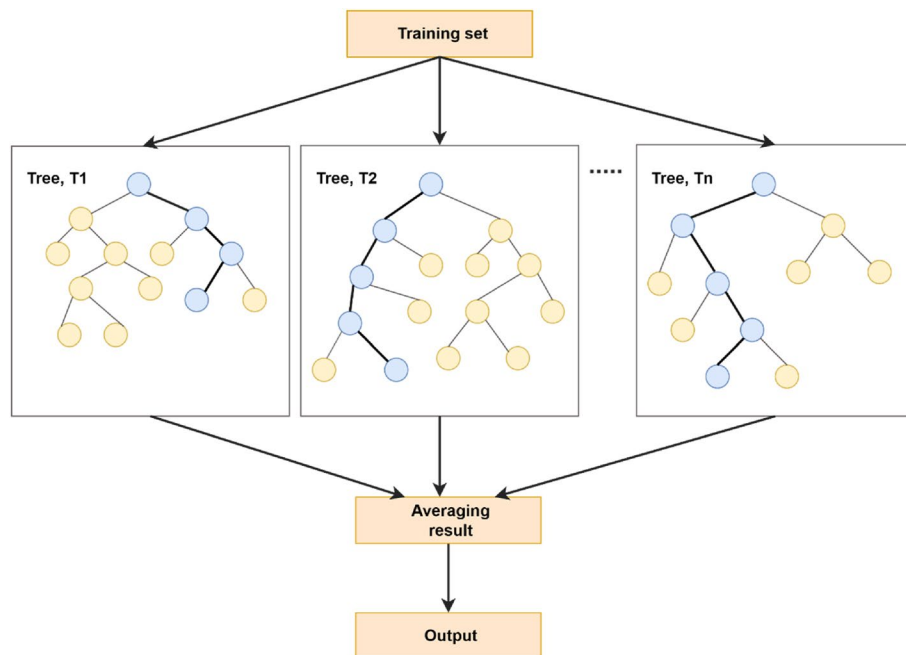
Here,  $R$  is the number of rows,  $m$  is the column number,  $y_i$  is the training value,  $y_j$  is the predicted value,  $\alpha$  is the hyperparameter, and  $a_k$  is the coefficient of the  $k$ -th feature.

### Balancing dataset

Improving ML accuracy greatly depends on balancing the imbalanced dataset [60]. An unbalanced dataset is one in which the number of observations for one of the target class labels is considerably smaller than for the other class labels [61]. Due to a lack of data, it will be challenging to obtain a meaningful and effective prediction model when a dataset is unbalanced or when a rare event happens [62, 63].

### Synthetic minority oversampling technique

SMOTE is commonly utilized to address class-imbalance issues in the healthcare industry [64]. To ensure a more even distribution of data, synthetic instances were produced rather than copied. Using the KNN algorithm, which serves as the foundation



**Fig. 2** Random Forest algorithm



of these techniques, distance computation between instances is essential to the process of creating synthetic samples. To generate a new sample, SMOTE first picks a set of instances that are relatively close to one another in the feature space, then draws a line connecting those examples, and finally, picks a point on that line [65].

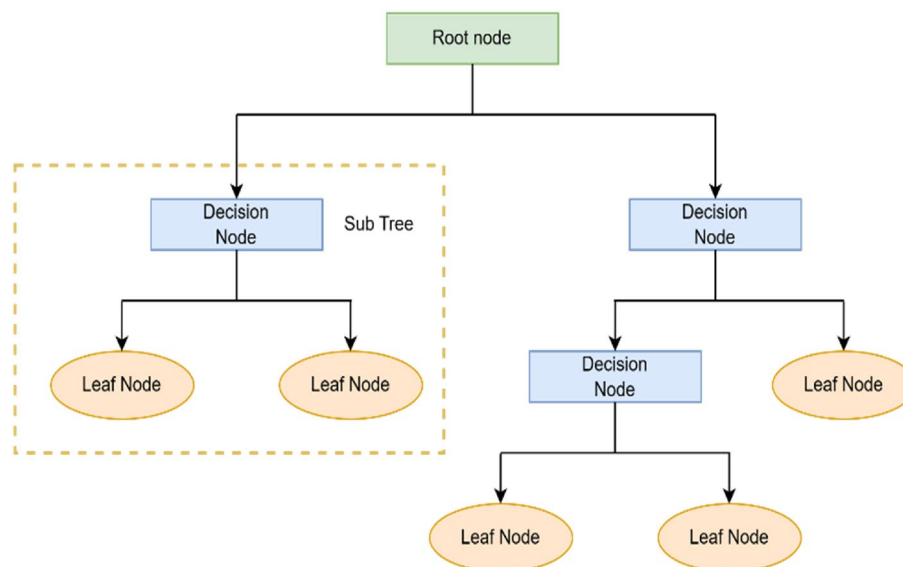
### Algorithms used

#### *Random forest (RF)*

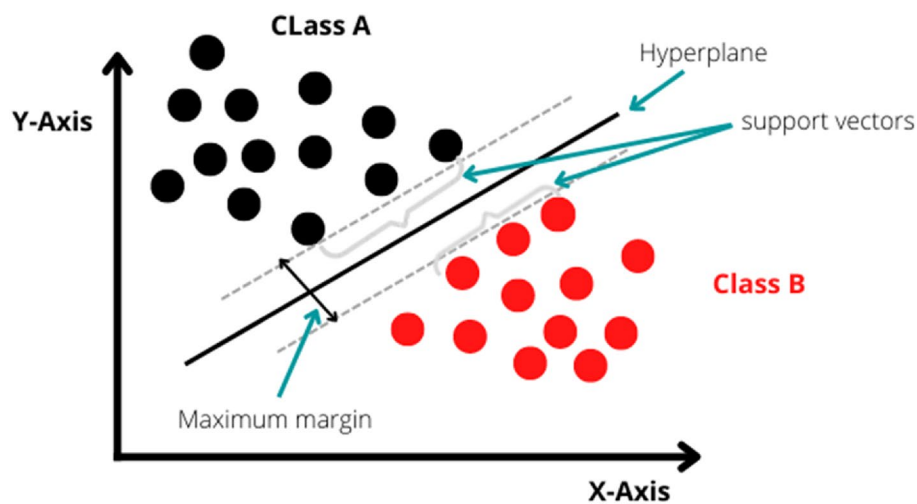
RF is characterized as an ensemble learner because it creates a significant number of classifiers and consolidates their outputs [66]. To increase the dataset's ability to predict outcomes, it uses a number of DTs on different subsets of the provided dataset and calculates the mean. Higher accuracy is obtained, and the overfitting issue is averted when there are more trees in the forest [67, 68]. Since the techniques use an amalgamation of trees to make their predictions about the dataset's class, some DTs might produce the correct result while others may not. But when all the trees are combined, they predict the right outcome [69–71]. The schematic diagram of the RF algorithm is shown in Fig. 2.

#### *Decision tree (DT)*

DT operates under the decision-making premise. The primary goal of employing DT in this study is to make predictions about the target class using a decision rule derived from historical data. It has a tree-like structure and is both precise and reliable. Any multistage method begins with the concept that a difficult decision may be broken down into a union of multiple smaller decisions, with the expectation that the resulting solution will be close to the desired solution [72]. In a DT, each possible branch is specified by a data-splitting sequence that starts at the root and ends with a Boolean result at the leaf node [73]. The training sample is represented by the initial node in the tree, which also contains internal nodes for dataset attributes, branching for decision-making processes,



**Fig. 3** Decision tree algorithm [75]



**Fig. 4** SVM algorithm

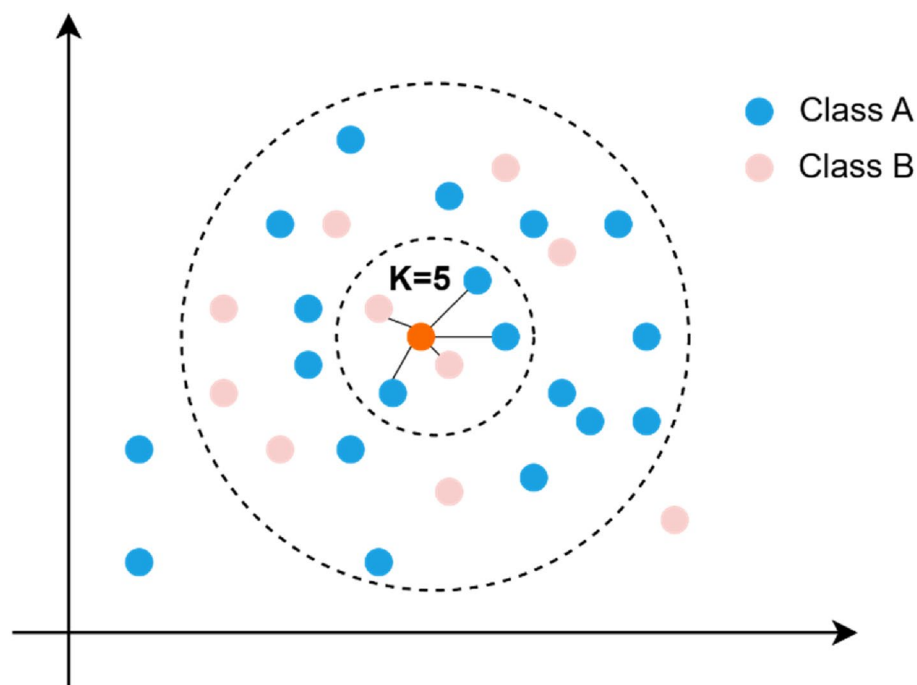
and leaf nodes for results [74, 75]. The schematic diagram of the DT algorithm is shown in Fig. 3.

#### **Support vector machine (SVM)**

SVM is a technique that examines data and categorizes it into one of two groups. It generates a representation of the sorted data, with the margins between both being as wide as feasible. Data points falling on one side of the line will be assigned to one category, while those falling on the other side will be assigned to another. It may discover intricate connections between your data without requiring much manipulation [76, 77]. The highest distance between the data points is what is referred to as a hyperplane, which is a decision boundary that contributes to classifying the data points. Support vectors are the number of observations or vectors nearest to a hyperplane that influences its location [78, 79]. The schematic diagram of the SVM is shown in Fig. 4.

#### **K-nearest neighbors (KNN)**

The KNN method determines which of "K" possible classes the test data most closely resembles [80] based on a probability calculation of how well each class fits the training data. By determining the difference between both the test data and all of the training points, KNN attempts to predict the proper class for the test data [81]. K represents the number of immediate neighbors. There are no pre-defined statistical procedures for determining the most advantageous value of K, although selecting a small value of K results in unstable decision boundaries [82, 83]. So, while utilizing the KNN method, the distance and K value are significant factors to take into account. It is essential to consider the input data when deciding on a number for k; data with more outliers or noise, for example, may fare better with larger values. In order to eliminate ties in categorization, it is often advised to choose an odd integer for k [84–86]. A representation of the KNN algorithm is shown in Fig. 5. Here, we can see that class A and class B instances are denoted by two different color. There is also an instance in orange color which is actually the new example or data that have to be classified by the algorithm.



**Fig. 5** KNN algorithm

#### ***Adaptive boosting (AB)***

AB may be used to enhance the effectiveness of any ML model. It works well with slow learners. The most effective algorithm for AB is DT for one level [87], making it the most widely employed. This approach initially trains the classification on the original dataset. The classifier is then trained on many instances, with each instance attempting to fix the flaw in the preceding instance. The classification algorithm is replicated, and each instance is trained on a unique data set. By giving weights to individual data points, several subsets of a dataset can be produced [88]. Then, a robust classifier is created by combining these weak classifiers with a cost function. In the final prediction, classifiers with greater accuracy are given a higher weight. The AB method can accept a parameter that specifies weak learners to which boosting should be performed [89, 90]. Our model's parameters for training were as follows: base estimator with DT, n-estimators values is 50, 0.5 learning rate, and 1 for the random state.

#### ***Gradient boosting (GB)***

GB is a boosting method for ML that represents a DT for large and complicated datasets. It combines several poor prediction models to produce a single robust model [91]. There are three main components to the GB process. The loss function varies depending on the task at hand, the weak predictors utilized, and the additive model, which adds trees using a gradient descent method [92]. By merging the next model with the preceding ones, the approach minimizes error while predicting the best model possible. By removing overfitting, GB is a method that can improve the efficiency of the algorithm. The overfitting impact is mitigated by regularization procedures. It also prevents degeneration when proper fitting processes have been performed. There is a positive

**Table 2** ML algorithms' training and prediction complexity

Algorithms	Training complexity	Prediction complexity
Random Forest	$O(n^2 p n_{\text{trees}})$	$O(p n_{\text{trees}})$
Support Vector Machine	$O(n^2 p + n^3)$	$O(p n_{\text{sv}})$
AdaBoost	$O(n p n_{\text{trees}})$	$O(p n_{\text{trees}})$
Decision Tree	$O(n^2 p)$	$O(p)$
Gradient Boosting	$O(n p n_{\text{trees}})$	$O(p n_{\text{trees}})$
K-Nearest Neighbors	N/A	$O(np)$

correlation between the number of GB rounds and the amount of error reduction achieved [93, 94]. Our model's best learning rate was 0.1.

### Computational complexity

Computational complexity is a discipline of computer science investigating algorithms based on the amount of computing power expected to run or execute them. Big O notation is the standard way to represent the time complexity of the algorithms [96]. KNN algorithms have no training complexity. Complexity is usually represented as an expression of  $n$ , where  $n$  is the size of the input,  $p$  is the number of attributes,  $n_{\text{trees}}$  represents the number of trees, and  $n_{\text{sv}}$  is the support vectors. The training and prediction complexity of the algorithms used in this study is given in Table 2.

### Evaluation metrics

Various measures, each with its unique significance, are used to evaluate each of the studies. When a prediction model generates an output that turns out to be true, we define that result as a true positive (TP). True Negative (TN) is the output of a prediction system that is false, and it is really false. False positive (FP) is the output of a predictive model that appears to be true but is in fact false. The result of a prediction model that is false but really correct is called a false negative (FN) [95].

**Accuracy:** It represents the proportion of input samples that resulted in accurate predictions.

$$\text{accuracy} = \frac{\text{Number of correct prediction}}{\text{total number of prediction}} \quad (2)$$

**Precision:** The precision determines the portion of valid positive predictions. It is calculated as the proportion of accurate positive findings to those that the classifier anticipated to be positive.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

**Recall:** It tends to compute the fraction of TP that was inaccurately classified. The ratio of TP to the combined total of TP and FN.

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

**Table 3** Different parameters of algorithms used in this study

Algorithm	Parameters value
KNN	n_neighbors = 5
RF	Default
DT	max_depth = 1, criterion = 'gini', random_state = 1
SVM	Default
AB	base_estimator = dtclf, n_estimators = 50, learning_rate = 0.5, algorithm = 'SAMME.R', random_state = 1
GB	learning_rate = 0.1

**Table 4** Classification result of all ML models using all features

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
KNN	94	63	80	67
RF	98	82	74	77
DT	97	77	71	74
SVM	76	53	73	50
AB	98	85	76	80
GB	98	80	80	80

**F1-score:** It involves assessing a binary classification model based on predictions for the positive class. Precision and Recall are employed in order to compute it. F1-score falls between 0 and 1. It indicates the classifier's accuracy, or how many examples it successfully classifies, as well as its robustness.

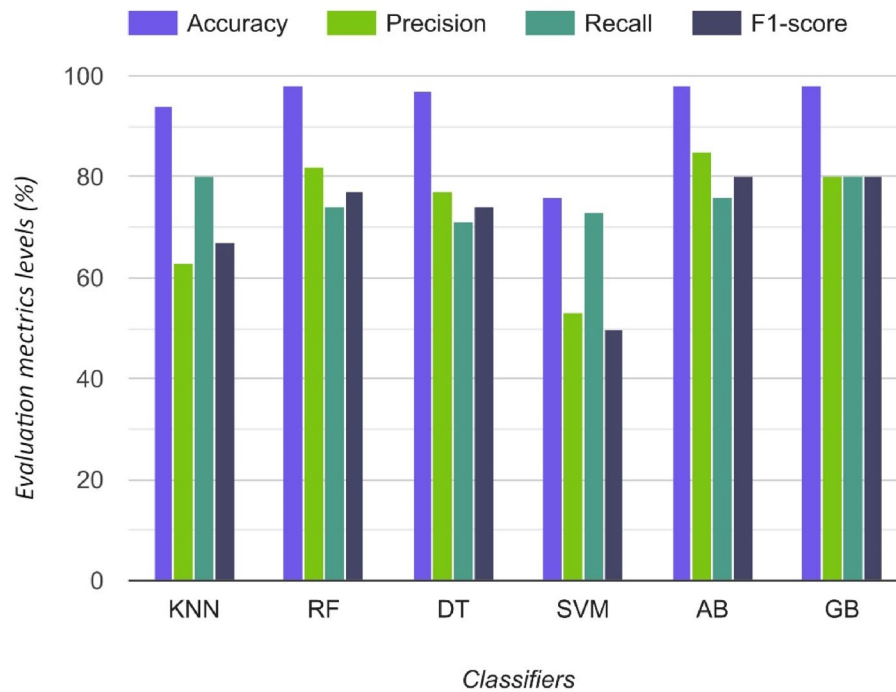
$$F1 - score = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (5)$$

## Result and discussion

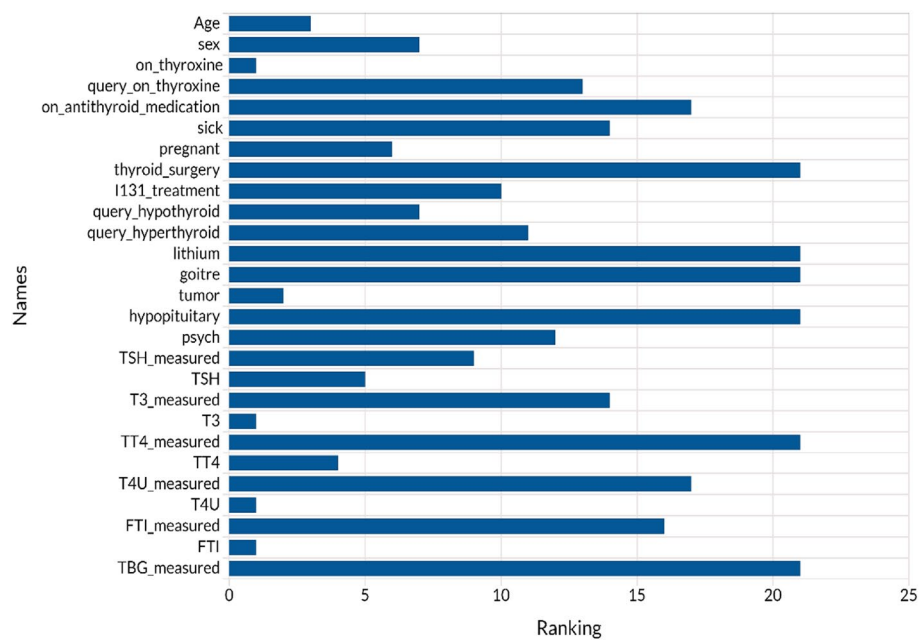
This section of the study consisted of the examination of classification models and their outputs from several distinct perspectives. In this study, we employed six ML algorithms: KNN, RF, DT, SVM, AB, and GB. These classification methods were applied to a number of parameters, which are depicted in Table 3 with a 5-fold CV. We first demonstrated the findings using all features, then the results using only the most important ones.

### Experimental results with all features

A comparison of ML classifiers is conducted on the complete set of features in our dataset. In terms of assessment measures, certain classifiers performed well, while others did not. This research has been implemented into tree-based ensemble models, such as DT and RF. Additionally, the AB and GB tree-based boosting models are also utilized. The performance assessment of ML models with their whole set of characteristics is shown in Table 4. Table 4 shows that with the exception of SVM, the



**Fig. 6** Comparisons graph of all features evaluation metrics

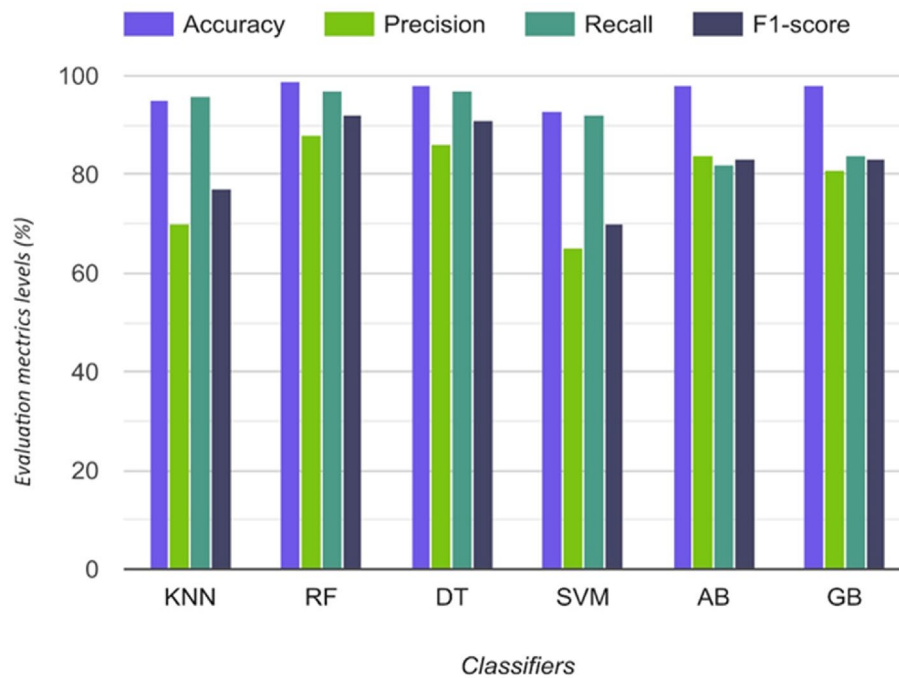


**Fig. 7** Boruta features ranking

accuracy of all methods is well and above 90%. The best efficiency is achieved with the GB classifier, which exhibits a 98% accuracy rate, 80% precision, 80% recall, and 80% F1-Score. Compared to the GB classifier, RF and AB algorithms achieve similar levels of accuracy; however, their other performance metrics values are slightly lower,

**Table 5** The performance metrics of the algorithms after applying Boruta (Fig. 8)

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
KNN	95	68	91	74
RF	99	96	90	93
DT	99	91	83	87
SVM	93	63	84	68
AB	98	84	78	80
GB	98	83	76	79

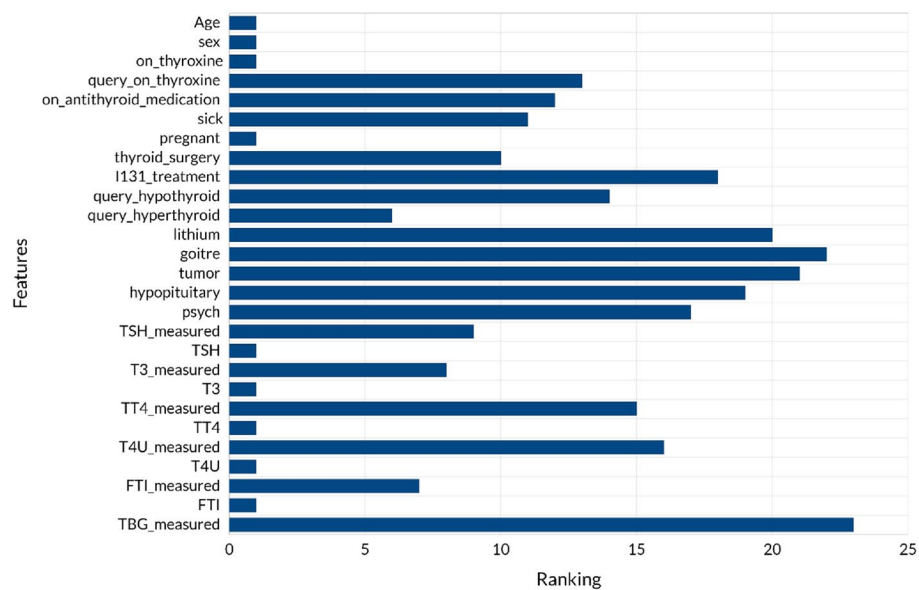
**Fig. 8** Comparison of different evaluation metrics using Boruta feature selection

with RF achieving a precision of 82%, recall of 74%, and f1-score value of 77%, and AB achieving a precision, recall, and f1-score of 85%, 76%, and 80%, respectively. Although AB and RF have better accuracy values than GB, GB's recall value is substantially greater. SVM has the weakest result, with 76% accuracy, 53% precision, 73% recall, and 50% f1-score (Fig. 6).

#### Experimental results with Boruta selected features

Four pertinent characteristics on thyroxin, T4U, FTI, and T3, are chosen from Fig. 7 based on the Boruta feature selection algorithm's ranking of them. Features with ranks greater than one are ignored by Boruta, and only those with a rank of one are considered. The effectiveness of these attributes is then examined using ML classifiers with the SMOTE approach. By eliminating the least significant features determined by Boruta, then the classifiers are trained and evaluated. After discarding unnecessary





**Fig. 9** RFE features ranking

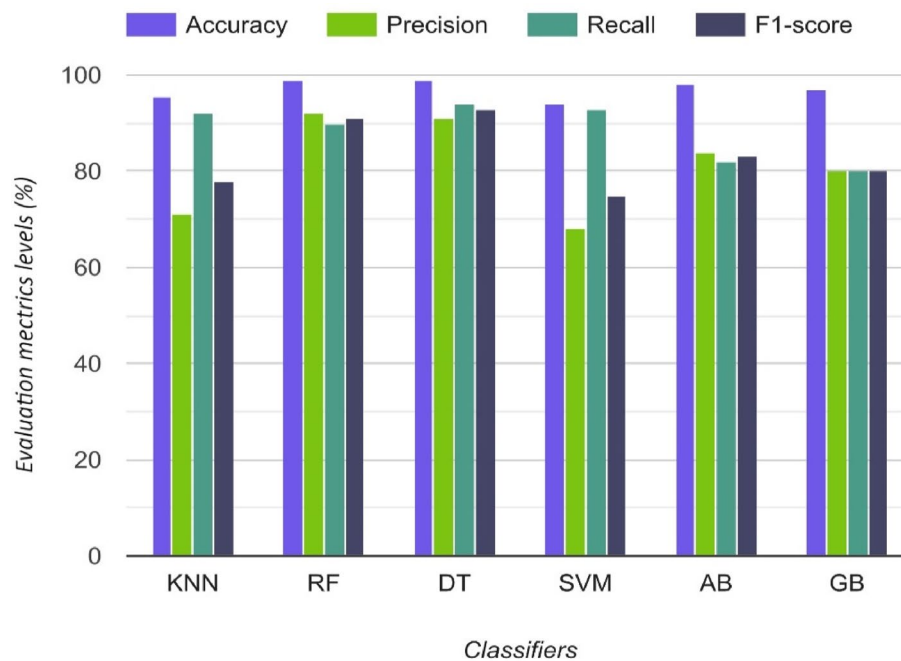
**Table 6** The performance metrics of the algorithms after applying RFE

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
KNN	97	76	96	83
RF	99	89	94	91
DT	99	88	92	90
SVM	93	62	91	68
AB	98	85	80	82
GB	98	83	85	84

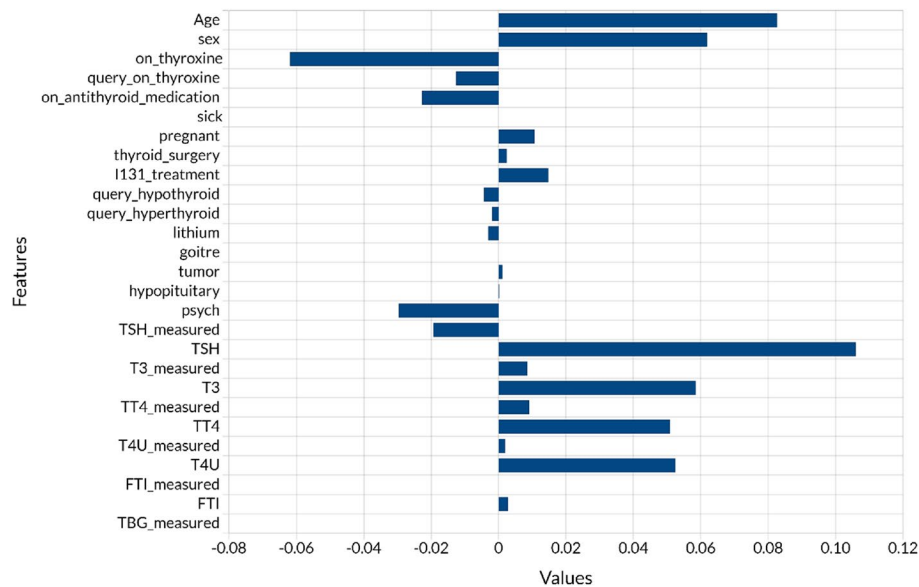
data, the classifiers' performance improved significantly. In Table 5, the accuracy outcomes after discarding the least significant features are shown. All algorithms' accuracy has improved greatly, with the exception of AB and GB, whose accuracy has remained constant at 98%. Significant gains are shown for both RF and DT when employing four characteristics, with RF's recall improving by 16%. Despite an improvement in KNN's overall performance values, its accuracy and f1-score value still persist below 80%. With these four characteristics, T4U, FTI, T3, and thyroxine, the RF algorithm stood out with an accuracy of 99%, precision of 96%, recall of 90%, and the f1-score of 93%, while DT performance metrics are quite close to RF. Again SVM performed poorly.

#### Experimental results with RFE

Figure 9 shows that using the RFE approach, the attributes are narrowed down to age, sex, on thyroxine, pregnant, TSH, TT4, T4U, T3, and FTI features with rank one, and then, six different types of ML classifiers are applied to those selected features. Table 6



**Fig. 10** Comparison of different evaluation metrics using RFE feature selection

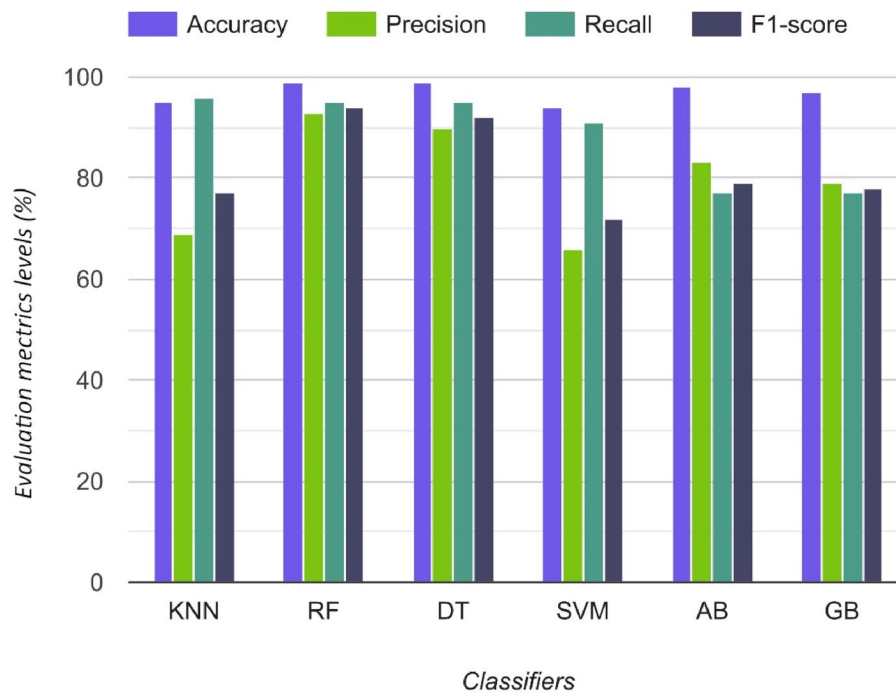


**Fig. 11** LASSO feature score

shows that with the exception of GB, the values of all performance measures for all algorithms enhanced, while the accuracy of the RF, DT, and AB classifiers persisted the same with the Boruta-selected four attributes. As can be seen, the RF classifier performed admirably with these nine characteristics, achieving accuracy rates of 99%, precision rates of 89%, recall rates of 94%, and f1-score rates of 91%. SVM, on the other hand, performs badly. The accuracy of the DT method is similar to that of the

**Table 7** The performance metrics of the algorithms after applying LASSO

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
KNN	96	74	91	80
RF	99	97	92	95
DT	99	95	92	94
SVM	94	66	91	73
AB	98	83	77	79
GB	97	79	77	78

**Fig. 12** Comparison of different evaluation metrics using LASSO feature selection

RF, while recall and f1-score values are 2% and 1% lesser than RF, respectively. The results for AB performance measures are almost the same as they were for four and all features, but the values for GB are higher than they were for Boruta-selected features. The KNN method has good accuracy, recall, and f1-score values, which were 97%, 96%, and 83%, respectively, apart from its precision value, which was 76% (Fig. 10).

#### Experimental results with LASSO selected features

Six features (age, sex, TSH, T3, TT4, and T4U) have the highest positive coefficient values, as seen in Fig. 11. Meanwhile, eight of the features have negative coefficient scores. With this approach, we are able to narrow down the characteristics to only six with the highest coefficient scores. Table 7 demonstrates the performance metrics value after applying LASSO method to the dataset. The accuracy values of all methods are exactly equivalent to those of the RFE-selected features. RF outperformed with the highest evaluation metrics values of 99% accuracy, 92% recall, 97% precision, and

95% f1-score. DT is the second most significant classifier, with a 99% success rate, 95% precision, 92% recall, and 94% f1-score. Despite SVM's outstanding accuracy of 94%, it has the lowest precision rating when compared to other methods. Without precision value, KNN performance metrics values are acceptable. In comparison with the features chosen by RFE, AB and GB performance worsened (Fig. 12).

It can be seen that all of the algorithms we used have a decent performance, DT and RF performed best with 99% accuracy, proving to be the most accurate in predicting TD. Since boosting methods are known to transform weak learners into strong learners, AB and GB performed well with all features, but as soon as we choose important attributes and use classification techniques, their performance begins to deteriorate. KNN and SVM precision values are the lowest compared to other algorithms in all stages, with all features, even with selected features. As both RF and DT algorithms are tree-based, the values of their performance measures are almost equal throughout all phases. Finally, after analyzing the outcomes of all features and selecting key features using Boruta, RFE, and LASSO, we found that RF with LASSO selected features achieved the best performance, with an accuracy of 99%, precision of 97%, recall of 92%, and an f1-score of 95%. The performance of DT was second to that of RF. Age, sex, TSH, T3, TT4, and T4U are considered to be the most significant attributes by LASSO. By including these features, our proposed approach received the

**Table 8** Comparison of outcome between the existing system and our proposed system

Authors	Algorithms	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	outcome
D.C Yadov et al. [33]	bagging, boosting, Stacking, and Voting	Chandan Diagnostic Center Sipah Jaunpur	94.210	94.412	96.620	N/A	Predict TD
L. Aversano et al. [34]	DT, NB, KNN, RF, EXT, MLP, XGB, CB, AB, GB	AOU Federico II	84	85	84	84	TD treatment prediction
T. Alyas et al. [35]	DT, RF, KNN, ANN	UCI	94.8	91	N/A	N/A	Predict TD
I.M.D. Maysanjaya et al. [37]	RBF, LVQ, MLP, BPA, AIRS, Perceptron	UCI	96.74	96.8	96.7	96.8	Predict TD
W. Ahmad et al. [38]	LDA, KNN, ANFIS	UCI	98.5	99.7	94.7	N/A	Predict TD
R. Chaganti et al. [39]	RF, LR, SVM, AD, and GB, LSTM, CNN	UCI	93	94	92	93	Predict TD
S. Shibu et al. [97]	KNN, RF, XGB, and CB	Kaggle	98.83	92.8	83.8	N/A	Predict TD
Our study	KNN, RF, DT, SVM, AB, and GB	UCI	99	97	92	95	Predict TD and identify major risk factors

AIRS: artificial immune recognition system, ANN: artificial neural networks, BPA: back propagation algorithm, CB: cat boost, CNN: convolutional neural network, EXT: extreme gradient boosting, LSTM: long short-term memory, LVQ: learning vector quantization, RBF: radial-based function,

highest scores possible on all evaluation metrics. As a result, these are the major risk factors for a person with this disease.

#### **Comparative table between the existing model and the proposed system**

In Table 8, the comparison of the outcome between this proposed framework and the previous studies has been presented. Here, it can be seen that the dataset description and the performance metrics such as accuracy, precision, recall, and F1-score were represented. The outcome of the studies was also shown in the table.

Finally, we have observed that the novel aspect of this study is the combination of robust scaling method, oversampling by SMOTE, and the use of feature selection methods to determine the best attributes that increase classification accuracy for thyroid disease prediction and identify major risk factors for thyroid disease.

#### **Conclusion and future scope**

In this research, we proposed a robust and effective ML-based method for predicting TD. KNN, RF, DT, SVM, AB, and GB are examples of ML approaches used for our study. SMOTE is implemented to address class imbalance issues. Additionally, the feature selection procedures RFE, Boruta, and LASSO are employed. Thus, experimental findings show that tree-based algorithms with LASSO technique selected features are particularly successful in reaching the best accuracy. On all assessment metrics, RF combined with LASSO performed best, with results of 99% accuracy, 97% precision, 92% recall, and 95% F1-score. Age, sex, TSH, T3, TT4, and T4U are considered to be the major risk factors for TD which are selected by LASSO.

This study has the possibility of improving the medical field and assisting as a helpful resource for doctors in identifying TD. Additionally, the doctor may benefit from having faster decision-making capabilities. However, the proposed model has certain shortcomings as well. The primary limitations of this study are the imbalanced dataset, the small sample size, and the substantial quantity of categorical variables compared to continuous features.

In the future, we intend to broaden the model's applicability so that it may be used with various feature selection techniques and be robust to datasets with significant amounts of missing data. Another potential approach is the use of DL algorithms and hybrid models. Making a web and mobile application for the prediction of TD and a self-monitoring system can add a certain value to the healthcare industry.

#### **Acknowledgements**

Not applicable

#### **Author contributions**

Both authors contributed equally in this work.

#### **Funding**

The authors received no financial support for the research authorship and publication of this article.

#### **Availability of data and materials**

The data used to support the findings of this study are available from the corresponding author upon request.

#### **Declarations**

##### **Competing interests**

The authors declare that they have no conflicts of interest.

Received: 22 February 2023 Accepted: 10 June 2023

Published online: 16 June 2023

**References**

- Brent GA (2012) Mechanisms of thyroid hormone action. *J Clin Invest* 122(9):3035–3043
- Boelaert K, Franklyn JA (2005) Thyroid hormone in health and disease. *J Endocrinol* 187(1):1–15
- Chen H-L, Yang B, Wang G, Liu J, Chen Y-D, Liu D-Y (2012) A three-stage expert system based on support vector machines for diagnosis. *J Med Syst* 36(3):1953–1963
- Tamer G, Arik S, Tamer I, Coksert D (2011) Relative vitamin D insufficiency in Hashimoto's thyroiditis. *Thyroid* 21(8):891–896
- Pearce EN, Farwell AP, Braverman LE (2003) Thyroiditis. *N Engl J Med* 348(26):2646–2655
- "General information/press room," *American Thyroid Association*, 13-Mar-2012. [Online]. Available: <https://www.thyroid.org/media-main/press-room/>. [Accessed: 16-Jan-2023].
- "Thyroid disease," *Cleveland Clinic*. [Online]. Available: <https://my.clevelandclinic.org/health/diseases/8541-thyroid-disease>. [Accessed: 16-Jan-2023].
- Stagnaro-Green A et al (2011) Guidelines of the American thyroid association for the diagnosis and management of thyroid disease during pregnancy and postpartum. *Thyroid* 21(10):1081–1125
- Zhang J, Lazar MA (2000) The mechanism of action of thyroid hormones. *Annu Rev Physiol* 62(1):439–466
- Vanderpump MPJ (2011) The epidemiology of thyroid disease. *Br Med Bull* 99(1):39–51
- Pearce EN, Andersson M, Zimmermann MB (2013) Global iodine nutrition: where do we stand in 2013? *Thyroid* 23(5):523–528
- Klein I, Danzi S (2007) Thyroid disease and the heart. *Circulation* 116(15):1725–1735
- Klein I, Ojamaa K (2001) Thyroid hormone and the cardiovascular system. *N Engl J Med* 344(7):501–509
- Schroeder AC, Privalsky ML (2014) Thyroid hormones, t3 and t4, in the brain. *Front Endocrinol (Lausanne)* 5:40
- Canaris GJ, Manowitz NR, Mayor G, Ridgway EC (2000) The Colorado thyroid disease prevalence study. *Arch Intern Med* 160(4):526–534
- Mortavazi S, Habib A, Ganj-Karami A, Samimi-Doost R, Pour-Abedi A, Babaie A (2009) Alterations in TSH and thyroid hormones following mobile phone use. *Oman Med J* 24(4):274–278
- Fazio S, Palmieri EA, Lombardi G, Biondi B (2004) Effects of thyroid hormone on the cardiovascular system. *Recent Prog Horm Res* 59(1):31–50
- Oppenheimer JH, Schwartz HL, Mariash CN, Kinlaw WB, Wong NC, Freaque HC (1987) Advances in our understanding of thyroid hormone action at the cellular level. *Endocr Rev* 8(3):288–308
- Farling PA (2000) Thyroid disease. *Br J Anaesth* 85(1):15–28
- Poppe K, Velkeniers B, Glinooer D (2007) Thyroid disease and female reproduction. *Clin Endocrinol (Oxf)* 66(3):309–321
- Mair C et al (2000) An investigation of machine learning based prediction systems. *J Syst Softw* 53(1):23–29
- Sarker IH (2021) Machine learning: Algorithms, real-world applications and research directions. *SN Comput Sci* 2(3):160
- Uddin S, Khan A, Hossain ME, Moni MA (2019) Comparing different supervised machine learning algorithms for disease prediction. *BMC Med Inform Decis Mak* 19(1):281
- Ghahramani Z (2015) Probabilistic machine learning and artificial intelligence. *Nature* 521(7553):452–459
- Horvitz E, Mulligan D (2015) Policy forum. Data, privacy, and the greater good. *Science* 349(6245):253–255
- Sebastiani F (2002) Machine learning in automated text categorization. *ACM Comput Surv* 34(1):1–47
- Joachims T (1998) Text categorization with Support Vector Machines: Learning with many relevant features. In: *Machine Learning: ECML-98, Berlin, Heidelberg: Springer Berlin Heidelberg*, pp. 137–142.
- Sommer R, Paxson V (2010) Outside the closed world: On using machine learning for network intrusion detection. In: *2010 IEEE Symposium on Security and Privacy*, 2010.
- Schmunk S, Höpken W, Fuchs M, Lexhagen M (2013) "Sentiment analysis: extracting decision-relevant knowledge from UGC", In: *Information and Communication Technologies in Tourism 2014*. Springer International Publishing, Cham, pp 253–265
- Eom J, Kim S, Zhang B (2008) AptaCDSS-E: a classifier ensemble-based clinical decision support system for cardiovascular disease level prediction. *Exp Syst Appl* 34(4):2465–2479
- Wang Y, Lamim Ribeiro JM, Tiwary P (2020) Machine learning approaches for analyzing and enhancing molecular dynamics simulations. *Curr Opin Struct Biol* 61:139–145
- Fy O et al (2017) Supervised machine learning algorithms: classification and comparison. *Int J Comput Trends Technol*, 48(3): 128–138
- Yadav DC, Pal S (2022) Thyroid prediction using ensemble data mining techniques. *Int J Inf Technol* 14(3):1273–1283
- Aversano L et al (2021) Thyroid Disease Treatment prediction with machine learning approaches. *Procedia Comput Sci* 192:1031–1040
- Alyas T, Hamid M, Alissa K, Faiz T, Tabassum N, Ahmad A (2022) Empirical method for thyroid disease classification using a machine learning approach. *Biomed Res Int* 2022:9809932
- Abbad Ur Rehman H, Lin C-Y, Mushtaq Z, Su S-F (2021) Performance analysis of machine learning algorithms for thyroid disease. *Arab J Sci Eng* 46(10):9437–9449
- Maysanjaya IMD, Nugroho HA, Setiawan NA (2015) A comparison of classification methods on diagnosis of thyroid diseases. In: *2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2015.
- Ahmad W, Ahmad A, Lu C, Khoso BA, Huang L (2018) A novel hybrid decision support system for thyroid disease forecasting. *Soft Comput* 22(16):5377–5383

39. Chaganti R, Rustam F, De La Torre Díez I, Mazón JLV, Rodríguez CL, Ashraf I (2022) Thyroid disease prediction using selective features and machine learning techniques. *Cancers (Basel)* 14(16):3914
40. "UCI machine learning repository: Thyroid disease data set," *Uci.edu*. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/thyroid+disease>. [Accessed: 18-Jan-2023].
41. Alexandropoulos S-AN, Kotsiantis SB, Vrahatis MN (2019) Data preprocessing in predictive data mining. *Knowl Eng Rev*, 34
42. García S, Luengo J, Herrera F (2016) Data preprocessing in data mining. Springer International Publishing, Cham, Switzerland
43. Liu N, Gao G, Liu G (2016) Data preprocessing based on partially supervised learning. In: Proceedings of the 6th International Conference on Information Engineering for Mechanics and Materials
44. Chen B (2023) Data collection and preprocessing. In: SpringerBriefs in Computer Science, Singapore: Springer Nature Singapore, 2023, pp. 5–16.
45. Kumar V (2023) Sklearn feature scaling with StandardScaler, MinMaxScaler, RobustScaler and MaxAbsScaler, MLK - Machine Learning Knowledge, 24-Jan-2022. [Online]. Available: <https://machinelearningknowledge.ai/sklearn-feature-scaling-with-standardscaler-minmaxscaler-robustscaler-and-maxabsscaler/>. [Accessed: 18-Jan-2023].
46. Cai J, Luo J, Wang S, Yang S (2018) Feature selection in machine learning: a new perspective. *Neurocomputing* 300:70–79
47. Hall MA (1999) Correlation-based feature selection for machine learning. The University of Waikato, Hamilton, New Zealand
48. Kursa MB, Rudnicki WR (2010) Feature Selection with the Boruta Package. *J Stat Softw*, 36, 2010.
49. Rudnicki WR, Wrzesień M, Paja W (2015) All relevant feature selection methods and applications," In: Feature Selection for Data and Pattern Recognition, Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 11–28.
50. Ali M (2023) Boruta feature selection explained in python. *Geek Culture*, 14-May-2022. [Online]. Available: <https://medium.com/geekculture/boruta-feature-selection-explained-in-python-7ae8bf4aa1e7>. [Accessed: 18-Jan-2023].
51. Kumar SS, Shaikh T (2017) Empirical evaluation of the performance of feature selection approaches on random forest," In: 2017 International Conference on Computer and Applications (ICCA), 2017.
52. Yan K, Zhang D (2015) Feature selection and analysis on correlated gas sensor data with recursive feature elimination. *Sens Actuators B Chem* 212:353–363
53. Chen X-W, Jeong JC (2007) Enhanced recursive feature elimination," In: Sixth International Conference on Machine Learning and Applications (ICMLA 2007), 2007.
54. Granitto PM, Furlanello C, Biasioli F, Gasperi F (2006) Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. *Chemometr Intell Lab Syst* 83(2):83–90
55. Fonti V, Belitser E (2017) Feature selection using lasso. *Curr Genomics* 30:1–25
56. Muthukrishnan R, Rohini R (2016) LASSO: a feature selection technique in predictive modeling for machine learning. In: 2016 IEEE International Conference on Advances in Computer Applications (ICACA)
57. Yamada M, Jitkritum W, Sigal L, Xing EP, Sugiyama M (2014) High-dimensional feature selection by feature-wise kernelized Lasso. *Neural Comput* 26(1):185–207
58. Zhou Y, Jin R, Hoi SCH (2010) Exclusive lasso for multi-task feature selection. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, PMLR 9:988–995
59. Smith A, Thakurta A (2013) Differentially private model selection via stability arguments and the robustness of the Lasso. In: Proceedings of the 26th Annual Conference on Learning Theory, PMLR, 2013, pp. 819–850.
60. Lemaitre G, Nogueira F, Aridas CK (2016) Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning," *arXiv [cs.LG]*.
61. Krawczyk B (2016) Learning from imbalanced data: open challenges and future directions. *Prog Artif Intell* 5(4):221–232
62. Chawla NV, Japkowicz N, Kotcz A (2004) Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explor* 6(1):1–6
63. Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G (2017) Learning from class-imbalanced data: Review of methods and applications. *Exp Syst Appl* 73:220–239
64. Mukherjee M, Khushi M (2021) SMOTE-ENC: A novel SMOTE-based method to generate synthetic data for nominal and continuous features. *Appl Syst Innov* 4(1):18
65. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2011) SMOTE: Synthetic minority over-sampling technique. *arXiv [cs.AI]*
66. Khalilia M, Chakraborty S, Popescu M (2011) Predicting disease risks from highly imbalanced data using random forest. *BMC Med Inform Decis Mak* 11(1):51
67. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
68. Seera M, Lim CP (2014) A hybrid intelligent system for medical data classification. *Expert Syst Appl* 41(5):2239–2249
69. Biau G (2010) Analysis of a random forests model. *arXiv [stat.ML]*
70. Pal M (2005) Random forest classifier for remote sensing classification. *Int J Remote Sens* 26(1):217–222
71. Oshiro TM, Perez PS, Baranauskas JA (2012) How many trees in a random forest?," In: Machine learning and data mining in pattern recognition, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 154–168.
72. Safavian SR, Landgrebe D (1991) A survey of decision tree classifier methodology. *IEEE Trans Syst Man Cybern* 21(3):660–674
73. Somvanshi M, Chavan P, Tambade S, Shinde SV (2016) A review of machine learning techniques using decision tree and support vector machine. In: 2016 International Conference on Computing Communication Control and automation (ICCCBEA), 2016.
74. Patel HH, Prajapati P (2018) Study and analysis of decision tree based classification algorithms. *Int J Comput Sci Eng* 6(10):74–78
75. Charbuty B, Abdulazeez A (2021) Classification based on decision tree algorithm for machine learning. *J Appl Sci Technol Trends* 2(01):20–28
76. Suthaharan S (2016) Machine learning models and algorithms for big data classification. Springer, US, Boston, MA



77. Awad M, Khanna R (2015) Efficient learning machines: Theories, concepts, and applications for engineers and system designers. Apress, Berkeley, CA
78. Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B (1998) Support vector machines. *IEEE Intell Syst* 13(4):18–28
79. Brereton RG, Lloyd GR (2010) Support vector machines for classification and regression. *Analyst* 135(2):230–267
80. Cunningham P, Delany SJ (2022) K-Nearest Neighbour classifiers - a tutorial. *ACM Comput Surv* 54(6):1–25
81. Zhang S, Cheng D, Deng Z, Zong M, Deng X (2018) A novel k NN algorithm with data-driven k parameter computation. *Pattern Recognit Lett* 109:44–54
82. Deng Z, Zhu X, Cheng D, Zong M, Zhang S (2016) Efficient kNN classification algorithm for big data. *Neurocomputing* 195:143–148
83. Guo G, Wang H, Bell D, Bi Y, Greer K (2003) KNN model-based approach in classification, In: *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 986–996.
84. Taunk K, De S, Verma S, Swetapadma A (2019) A brief review of nearest neighbor algorithm for learning and classification," In: *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019.
85. Zhang S, Li X, Zong M, Zhu X, Wang R (2018) Efficient kNN classification with different numbers of nearest neighbors. *IEEE Trans Neural Netw Learn Syst* 29(5):1774–1785
86. Zhang S, Li X, Zong M, Zhu X, Cheng D (2017) Learning  $k$  for kNN classification. *ACM Trans Intell Syst Technol* 8(3):1–19
87. Rätsch G, Onoda T, Müller K-R (2001) Soft Margins for AdaBoost. *Mach Learn* 42(3):287–320
88. Schapire RE (2013) Explaining AdaBoost, In: *Empirical Inference*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 37–52.
89. Schapire RE (2003) The boosting approach to machine learning: An overview, In: *Nonlinear Estimation and Classification*, New York, NY: Springer New York, pp. 149–171.
90. Dietterich TG (2000) Ensemble Methods in Machine Learning," In: *Multiple Classifier Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–15.
91. A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Front. Neurobot.*, vol. 7, 2013.
92. Friedman JH (2002) Stochastic gradient boosting. *Comput Stat Data Anal* 38(4):367–378
93. Bentéjac C, Csörgő A, Martínez-Muñoz G (2021) A comparative analysis of gradient boosting algorithms. *Artif Intell Rev* 54(3):1937–1967
94. Binder H, Gefeller O, Schmid M, Mayr A (2014) The evolution of boosting algorithms: from machine learning to statistical modelling. *Methods Inf Med* 53(06):419–427
95. Japkowicz N, Shah M (2015) "Performance evaluation in machine learning", In: *Machine learning in radiation oncology*. Springer International Publishing, Cham, pp 41–56
96. Arora S, Barak B (2012) Computational complexity: a modern approach. Cambridge University Press, Cambridge, England
97. Shibu S, Sahu D (2023) Improvisation of predictive modeling using different classifiers for predicting thyroid disease in patients," pp. 1–11, doi: [https://doi.org/10.1007/978-3-031-31153-6\\_1](https://doi.org/10.1007/978-3-031-31153-6_1).

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)