Thyroid Disease Classification Based on Blood Tests Using Machine Learning Techniques

Michalina Sierko, Sandra Śmigiel and Damian Ledziński

June 10, 2024

Michalina SIERKO[1,*], Sandra ŚMIGIEL[1], Damian LEDZIŃSKI[1]

1. THYROID DISEASE CLASSIFICATION BASED ON BLOOD TESTS USING MACHINE LEARNING TECHNIQUES

1.1. Introduction

Machine learning and artificial intelligence are becoming crucial tools in the field of medicine. One area of interest is the thyroid gland, which produces and regulates many essential functions in the body. The thyroid gland is responsible for accumulating iodine from the blood, which it then uses to produce thyroid hormones T4 and T3. Thyroid diseases are a common health issue affecting millions of people worldwide, and the number of individuals impacted continues to rise each year. The diagnosis and monitoring of thyroid diseases have traditionally relied on biochemical analysis, including blood tests, which provide vital information about the condition of this gland [5, 8].

In recent decades, there has been a dynamic development of information technologies that have significantly transformed many areas of life, including medicine. This phenomenon is particularly evident in the use of machine learning and artificial intelligence methods, which enable the analysis of large datasets. Machine learning is becoming a promising tool that can lead to significant advancements in the diagnosis of thyroid diseases. The application of machine learning models in analyzing blood test results can result in more precise classification of thyroid diseases. This is especially important in the context of personalized medicine, where accurate diagnosis and prompt intervention can greatly improve patients' quality of life.

The focus has been on investigating the potential of machine learning in the context of analyzing blood test results and diagnosing thyroid diseases, emphasizing the

[1] Bydgoszcz University of Science and Technology, Bydgoszcz, Poland. Corresponding author: micsie001@pbs.edu.pl.

importance of integrating modern technologies with traditional medical methods. The primary aim of this research was to classify thyroid diseases based on blood test results using machine learning techniques. The application of these innovative methods can not only improve the accuracy of diagnoses but also contribute to a better understanding of the pathogenic mechanisms involved.

Initially, an analysis of the literature was presented, highlighting the results and achievements of other researchers who have addressed this topic. Following this, the adopted research methodology was discussed, including a description of the selected database, the classification models used, and the available classification evaluation metrics.

1.2. Review of literature

In this section, an analysis of the literature related to the research topic was conducted. The focus was on key classification methods, algorithms, and the latest advancements in the field of classification, based on databases concerning thyroid diseases.

Table 1

Selected articles addressing the research problemental models

Position	Search Engine	Title	Date	Authors	Methods
1	Springer	A comparative study on thyroid disease detection using K-nearest neighbor and Naive Bayes classification techniques	2017	Khushboo C., Veenita K., Sai S., Tanupriya C., Saurabh M.	KNeighbors, Naive Bayes
2	Google Scholar	Comparison of Decision Tree, Neural Network, Statistic Learning, and k-NN Algorithms in Data Mining of Thyroid Disease Datasets	2018	Wafaa A.S., Riyad A. S.	MLP, J48, LMT, BayesNet, Naive Bayes, SMO, IBk, Random Forest
3	Google Scholar	Thyroid Disease Prediction Using Hybrid Machine Learning Techniques: An Effective Framework	2020	Yasir I. M., Dr. Sonu M.	SVM, Naive Bayers, J48, Bagging, Boosting, Decision Tree

4	Google Scholar	Thyroid Disease Multi-class Classification based on Optimized Gradient Boosting Model	2023	Alnaggar M., Handosa M., Medhat T., Rashad M. Z.	XGBoost
5	Springer	Estimation of Thyroid by Means of Machine Learning and Feature Selection Methods	2023	Dhamodaran S., Shankar B. B., Balachander B., Saravanan D., Kharate D.S.	KNeighbors, Naïve Bayes, SVM, ESTDD
6	Springer	An Explainable Artificial Intelligence Framework for the Predictive Analysis of Hypo and Hyper Thyroidism Using Machine Learning Algorithms	2023	Hossain B., Shama A., Adhikary A., Raha A.D., Aslam Uddin K. M., Hossain M.A., Islam I., Murad S.A., Munir S., Bairagi A.K.	Decision tree, Random Forest Classifier, XGBoost, Naive Bayes Classifier, KNeighbors, Logistic Regression, SVM

The first publication focuses on comparing methods for detecting thyroid diseases using machine learning techniques. Two learning models were employed: KNeighbors and Naive Bayes. The publication was based on data from the Knowledge Extraction Evolutionary Learning (KEEL) website, which included 7,200 patients and 21 classes. According to the article, the first mentioned machine learning method achieved a classification accuracy of 93.44%, while Naive Bayes only reached 22.56%. The study also utilized the Kappa parameter, which yielded results of 0.199 and 0.044, respectively [2].

The article published in 2018, like the others described, pertains to the classification of thyroid diseases based on blood test results. The dataset was sourced from the UCI Machine Learning repository and includes 21 attributes and 7,200 cases, divided into a training set with 3,772 cases and a test set with 3,428 cases. The study utilized machine learning models listed in Table 2 and evaluation metrics such as ACC, Kappa, MCC, and ROC. All ACC results exceeded the 92% threshold, with the best performances achieved by J48 trees at 99.4%, and LMT trees and Random Forest both at 99.2% [7].

The dataset from the study presented in position 3 of Table 1 originates from the main dataset collected at Sawai Man Singh Hospital in Jaipur, India. The study employed six evaluation metrics: ACC, Specificity, Sensitivity, Precision, Recall, and ROC to assess various machine learning methods. The SVM classifier proved to be the best, outperforming other proposed models in most metrics except Sensitivity and ROC. SVM achieved an ACC of 99.09%, with Precision, Recall, and Specificity all scoring 0.991 [6].

The study from position 4 of Table 1 was based on a dataset concerning thyroid diseases developed by the Garvan Institute, obtained from the UCI Machine Learning repository. The authors of this study aimed to optimize the hyperparameters of the XGBoost model and compare the obtained result with previous research findings on the same dataset. In this case, an ACC of 99% was achieved, demonstrating that the application of this method provides more accurate results compared to the same algorithms without utilizing hyperparameter optimization in machine learning models [1].

The article listed in position 5 and the research conducted therein also relied on the dataset presented, among others, in position 3 of the same table. The authors of this work indicated the use of three machine learning models: Naive Bayes, KNeighbors, and SVM. Their goal was to compare the effectiveness of different algorithms in predicting thyroid diseases. Additionally, they proposed the ESTDD model, which proved to be the most accurate. Based on the reported results, it can be observed that the proposed ESTDD model achieved ACC levels of 98.53%, Recall of 97.23%, and Sensitivity of 99.23% [3].

The last entry in Table 1, in its research, also relied on the dataset contained in the UCI Machine Learning repository. Seven different models presented in the table were utilized in the study. Evaluation of the models was conducted using metrics such as ACC, Precision, Recall, and F1-score. Regarding feature importance method, the highest ACC results were achieved by the Random Forest model at 91.42%, followed by Decision Tree at 90.5%, and XGBoost at 90.43%. However, with the univariate feature selection technique, Random Forest also provided the highest result at 90.4%, followed by Decision Tree at 89.55%, and XGBoost at 89.35% [4].

1.3. Materials and Mthods

In this section, a two-stage classification strategy was presented. The overall research process is depicted in Figure 1. For the purpose of the study, 8 steps were carried out to select the best machine learning models, followed by their optimization.

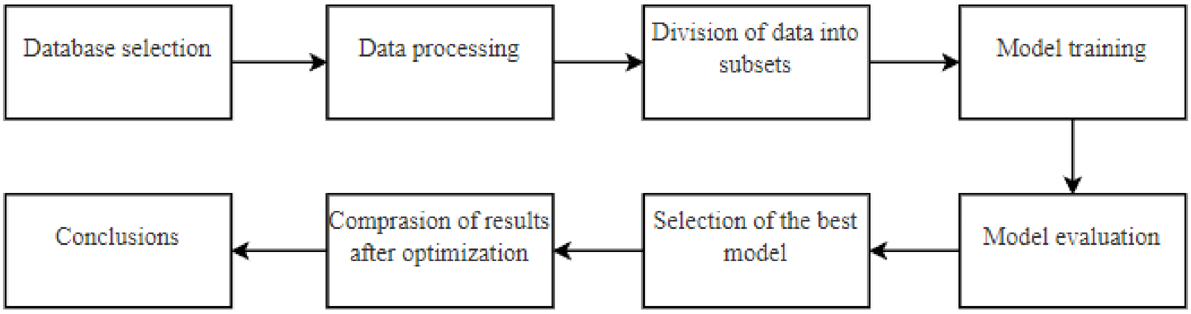
Database selection
Data processing
Division of data into subsets
Model training
Model evaluation
Selection of the best model
Comprasion of results after optimization
Conclusions

Fig. 1. The complete workfow paradigm in this study
Rys. 1.Ogólny schemat badań

The subject database was created by integrating datasets related to thyroid diseases obtained from the UCI Machine Learning repository. The Thyroid Disease Data [12] database was available on the Kaggle.com platform. It comprised 9172 records and 31 columns, containing metadata such as:

1) Demographic data,
2) Co-existing conditions of patients,
3) Information regarding administered substances,
4) Details about blood test measurements and their values,
5) And other relevant information.

The demographic data included the columns: age and sex. Meanwhile, the columns related to co-existing conditions were: sick, thyroid_surgery, query_, query_hyperthyroid, goitre, tumor, hypopituitary, psych, and pregnant. The columns on_thyroxine, query_on_thyroxine, I131_treatment, on_antithyroid_meds, and lithium provided information about substances taken by patients. Details about blood test measurements and their values were indicated by the columns: TSH_measured, TSH, T3_measured, T3, TT4_measured, TT4, T4U_measured, T4U, FTI_measured, FTI, TBG_measured, and TBG. The last category was characterized by the columns: referral_source, target, and patient_id.

The initial data preprocessing involved data preparation and normalization. This processing aimed to prepare the data for further analysis, including data cleaning by removing errors and rows with empty cells, which is essential for the proper machine learning process. Normalization served as a standardization process, facilitating data interpretation and analysis, which included tasks such as standardizing text and format and removing special characters.

Next, the data was divided into training data, which was used to train the model, and test data, on which the trained model was evaluated, among other things, in terms of effectiveness. The training data comprised a larger set containing all the relevant dependencies between the input and output data, allowing the model to properly learn these relationships. The most important feature of the test data was that it did not overlap with the training data, and the model evaluation was conducted based on data that it did not learn during training.

Model training involved adjusting the model parameters to the training data. The methodology of training primarily depended on the selected machine learning model. The study utilized models such as Decision Tree, Random Forest, XGBoost, LGBM, KNeighbors, MLP, SVC, and SGDC.

Model evaluation was based on assessing how well the model performed with the test data. Metrics such as ACC, F1-score, Precision, Recall, and BACC were used for this evaluation.

After evaluating all the models, the selection process identified those that performed best after checking the evaluation metric results. Three models stood out as the top performers: Random Forest, XGBoost, and LGBM.

In the case of the Random Forest model, the parameters n_estimators, min_samples_split, and min_samples_leaf were optimized. N_estimators represents the number of trees in the random forest. Its default value is 100. The next parameter, min_sample_split, with a default value of 2, represents the minimum number of samples required to split an internal node, while min_samples_leaf, with a standard value of 1, represents the minimum number of samples that must be present in a leaf node for it to be created. It is likely that a split point at any depth was only considered if min_samples_leaf left at least a specified number of training samples in each left and right branch. This may have had an effect on smoothing the model, especially in the regression process [11].

In the XGBoost model, the parameters n_estimators, min_samples_leaf, and max_depth were selected. The first of these parameters represents the number of boosting rounds to perform. Gradient boosting is quite robust to overfitting, so a large number usually results in better performance. This value must be between 1 and infinity. The second of these parameters, which has a default value of 1, represents the minimum number of samples that must be present in a leaf node. At any depth, a split point is only considered if min_samples_leaf leaves at least a minimum number of training samples in each left and right branch. The max_depth parameter, with a default value of 3, specifies the maximum depth of each regression estimator. Its purpose is to limit the number of nodes in the tree. It allows for the best possible performance [10].

In the case of the last selected classifier, namely LGBM, four parameters were optimized: num_leaves, n_estimators, subsample_for_bin, and min_child_samples. The first of these parameters represents the maximum number of tree leaves, with a default value of 31. The second of these parameters represents the number of boosted trees to fit, with a value of 100. The subsample_for_bin parameter, with a default value of 200,000, specifies the number of samples, and the last of these parameters, min_child_samples, with a default value of 20, represents the minimum amount of data required in a single leaf [9].

In addition, the default values of the models were compared with the values that were changed during the optimization process. An analysis of three evaluation metrics, namely ACC, F1-score, and BACC, was also conducted for the machine learning models, before and after optimization.

1.4. Results

This section presents the results for each machine learning model. In addition, the results of the evaluation metrics for all applied machine learning models were compared.

The three best models were optimized to improve their performance, which involved selecting the appropriate parameters for these models. In the context of the evaluation metrics, namely ACC, F1-score, and BACC, the results before and after parameter selection were compared.

1.4.1. Results for the selected machine learning models

Table 2 presents the results of five evaluation metrics, namely ACC, F1-score, Precision, Recall, and BACC, for the applied machine learning models. The best results for each metric are highlighted in bold in each row.

Table 2

Results for all applied machine learning models

Machine learning models Evaluation metrics	LGBM	XGBoost	Random Forest	Decision Tree	MLP	Kneighbors	SGDC	SVC
ACC	**0,955**	0,953	0,951	0,942	0,878	0,819	0,801	0,791

F1-score	**0,880**	0,871	0,879	0,841	0,658	0,422	0,326	0,238
Presision	0,877	**0,884**	0,883	0,853	0,616	0,345	0,257	0,203
Recall	**0,886**	0,861	0,878	0,832	0,744	0,673	0,540	0,356
BACC	0,886	0,861	0,878	0,832	0,744	0,673	**0,899**	0,889

Tree-based models achieved the highest results in the context of classification, demonstrating effectiveness above 94% for the ACC accuracy measure, 84% for the F1-score, and Precision and Recall exceeding 85% and 83%, respectively. SGDC and SVC, however, proved to be the most effective in terms of balanced accuracy BACC, exceeding the 88.9% threshold. However, comparing all the results obtained, LGBM, XGBoost, and Random Forest models turned out to be the best in the context of thyroid disease classification.

1.4.2. Machine Learning Model Optimization Results

Optimization of the top three classification models was conducted as part of the study. This optimization aimed to enhance the evaluation metrics ACC, F1, and BACC for the LGBM, XGBoost, and Random Forest models. It involved selecting appropriate model parameters tailored to each of the aforementioned models. The achieved improvement in classification metrics suggested that model optimization could be an effective approach to enhancing their performance.

A significant improvement in model effectiveness was observed after optimization compared to the initial results, as presented in Table 3. The models were sorted by the ACC parameter from highest to lowest.

Table 3

Comparison of Machine Learning Model Results Before and After Optimization

	Before optimization			After optimization		
Model	LGBM	XGBoost	Random Forest	LGBM	XGBoost	Random Forest
ACC	0,955	0,953	0,951	0,955	0,949	0,954
F1	0,88	0,871	0,879	0,955	0,949	0,954
BACC	0,886	0,881	0,878	0,877	0,8795	0,878

Table 4

Comparison of Machine Learning Model Parameters Used for Optimization

	Parameters	Default values	Optimized values
LGBM	num_leaves	31	32
	n_estimators	100	100
	subsample_for_bin	200000	0,1
	min_child_samples	20	2
XGBoost	n_estimators	100	200
	min_samples_leaf	1	2
	max_depth	3	4
Random Forest	n_estimators	100	500
	min_samples_split	2	5
	min_samples_leaf	1	2

Parameter optimization results for the LGBM model showed minimal changes in model performance metrics. The parameters presented in Table 4 were modified, but the results of metrics such as ACC, F1, and BACC remained almost constant. The ACC metric value remained at 95.5%, which may indicate that parameter optimization does not have a key impact on overall classification accuracy. It is worth noting that the largest change was achieved for the F1-score metric, where the value increased by 0.225. As mentioned earlier, F1-score takes into account both precision and recall, suggesting that parameter optimization contributed to improving the balance between these two measures. This was likely due to a more optimal configuration, which in turn improved the model's ability to simultaneously consider both positive and negative cases. The BACC value decreased slightly by 0.009, which may indicate that parameter optimization affected the balance of classification effectiveness between classes, but this change was not significant. In the context of LGBM model parameter optimization, it can be assumed that the num_leaves and n_estimators parameters had potentially a greater impact on metric evaluation results than the subsample_for_bin and min_child_samples parameters. Increasing the number of leaves could potentially increase the ability of the machine learning model to fit the training data. The parameter defining the number of samples, analyzing Table 6, may have had a minimal impact, as even after reducing it to 0.00005% of the default value, the result of the machine learning process remained almost unchanged. The last of the mentioned parameters and its optimization could have affected the control of tree depth and prevented overfitting by requiring a larger number of samples in a node. In summary, after optimizing the parameters of the LGBM model, minimal changes in metric results indicate that default values may be equally effective and subtle modifications had a limited impact on the effectiveness of the discussed classification model.

Changes in metric results were observed after optimizing the parameters of the XGBoost model. The parameter modification slightly lowered ACC from 0.953 to 0.949, which may suggest that adjusting some parameters affected the overall classification accuracy. The F1-score metric value once again showed the best improvement, as it increased by 0.078, which may indicate a better balance between precision and recall, as well as a reduction in model overfitting. In the case of BACC, a slight decrease was observed, as the difference was less than 0.002. Increasing n_estimators may have introduced additional complexity, but in this case, it did not lead to significant changes. Probably, the second optimized parameter had the greatest impact on the machine learning process. It could have helped to avoid overfitting and additionally better balance precision and recall, which resulted in better F1-score results. Analyzing the last of the parameters, more precisely its increase, could have led to increased model complexity. Therefore, changes in the min_samples_leaf parameters probably had the greatest impact on improving the F1-score measure, suggesting that optimizing this parameter was crucial for optimizing the XGBoost model. In the case of n_estimators and max_depth, the changes were less significant, so the default values were sufficiently effective. Such observations may suggest that the default values for selected parameters of a given model may be sufficiently effective.

Parameter optimization results for the Random Forest model showed improvements in all metrics except for balanced accuracy. The optimization led to an increase in ACC from 0.951 to 0.954, suggesting that the model is better at correctly classifying observations. This could be due to the overall increase in the number of trees and the more restrictive splitting conditions. The largest improvement was once again observed for the F1-score metric, which increased by almost 0.08. The increase in F1-score may indicate that the model is better at balancing precision and recall, which was important in the context of imbalanced classes in the dataset, and this may have caused BACC to remain stable. Increasing the first optimized parameter, the number of trees, likely had a significant impact on the learning process, as a larger number of trees could lead to an improvement in, for example, the ACC metric, due to the increased diversity of the model. It is worth noting that each decision tree brings a unique perspective, which may indicate a better ability to capture the complexity of the data used. The larger minimum number of samples required to split nodes made the decision tree more rigorous in creating split rules. This likely led to more stable and general rules, which in turn had a positive impact on the ability to generalize to new data. Increasing the min_samples_leaf parameter may have made the Random Forest model less prone to focusing on individual cases in the dataset, causing it to focus more on representative splits. Therefore, it can be assumed that this parameter could have acted as a mechanism to control model complexity, resulting in less susceptibility to overfitting. In summary, the optimization of the selected parameters proved to be very effective, as improvements were observed in two key evaluation metrics, and one maintained a stable level, which indicates the effectiveness of the applied optimization.

Compared to LGBM and XGBoost, Random Forest achieved the best results in optimization. While both other models, LGBM and XGBoost, also showed some improvement, Random Forest stood out as the most effective and efficient classification model in the applied context.

1.5. Conclusion

In the comparative analysis of this study with the results described in the literature review section, common characteristics were observed across all studies. The conducted study showed inferior results compared to the four discussed scientific articles. However, in the publication listed sixth in Table 1, lower effectiveness was achieved, with an ACC metric at the level of 91.42% for the Random Forest model. In the conducted studies, the same model achieved 95.1% effectiveness before optimization and 95.4% after it. Additionally, for the KNeighbors model, better results were obtained than those presented in the study placed first in Table 1. For the remaining articles, the accuracy of models, also used in this work, exceeded 98% for the ACC metric. The higher results of other publications may have resulted from a smaller number of patients and diagnosed diseases compared to the conducted study. In a situation where more classes and data were used in the classification process, this process could have become more complicated, potentially affecting the lower effectiveness of the evaluated metrics.

In summary, the expansion of research on thyroid disease classification should include refining models, studying their effectiveness in various populations, integrating them with clinical and imaging data, and considering ethical aspects related to new data sources. Striving for a more comprehensive and precise diagnosis through advanced machine learning techniques represents a significant step towards improving care for patients with thyroid diseases.

Bibliography

1. M. Alnaggar, M. Handosa, T. Medhat., M.Z., Thyroid Disease Multi-class Classification based on Optimized Gradient Boosting Model (2023) 2(1): 1-13.
2. K. Chandel, V. Kunwar, S. Sabitha, *A comparative study on thyroid disease detection using K-nearest neighbor and Naive Bayes classification techniques* (2016) **4**: 313-319.
3. S. Dhamodaran, B.B. Shankar, B. Balachander, D. Saravanan, D.S. Kharate, *Estimation of Thyroid by Means of Machine Learning and Feature Selection Methods* (2023).
4. M.B. Hossain, A. Shama, A. Adhikary, *An Explainable Artificial Intelligence Framework for the Predictive Analysis of Hypo and Hyper Thyroidism Using Machine Learning Algorithms* (2023) **3**: 211-231.
5. S.J. Konturek, *Fizjologia człowieka* (2007): 778-786.
6. Y.I. Mir, S. Mittal, *Thyroid Disease Prediction Using Hybrid Machine Learning Techniques: An Effective Framework* (2020) **9(2)**.
7. W. Somali, R. Shammari, *Comparison of Decision Tree, Neural Network, Statistic Learning, and k-NN Algorithms in Data Mining of Thyroid Disease Datasets* (2018) **5**: 241-246.
8. M. Szpinda, *Anatomia Prawidłowa Człowieka* (2022) **3**: 216-217.
9. https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html.
10. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html.
11. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.
12. https://www.kaggle.com/datasets/emmanuelfwerr/thyroid-disease-data.

CLASSIFICATION OF THYROID DISEASES BASED ON BLOOD TESTS USING MACHINE LEARNING TECHNIQUES

KLASYFIKACJA CHORÓB TARCZYCY NA PODSTAWIE WYNIKÓW BADAŃ KRWI PRZY POMOCY UCZENIA MASZYNOWEGO

Abstract

The thyroid gland is an important organ in the human body responsible for secreting hormones that support and stabilize the metabolic cycle. The dynamic development of technology and progress in monitoring thyroid diseases constitute a rapidly evolving area of research. The aim of the study was to classify thyroid diseases based on blood test results using machine learning techniques. Literature analysis, anatomical-physiological aspects of the thyroid, and advanced technologies such as machine learning were presented as powerful tools in monitoring and diagnosing thyroid disorders. A literature analysis was conducted based on six articles from the machine learning and thyroid disease classification domain. Laboratory diagnostics based on blood tests, as well as statistics and epidemiological data shaping the understanding of existing thyroid-related issues, were presented. The classification stage described the database, its content, and size. The tools, technology, and learning models used were also presented and discussed. LGBM, XGBoost, and Random Forest models were highlighted. Subsequently, optimization of these three models was performed. The Random Forest model provided the highest accuracy, F1-score and BACC. Classifying thyroid diseases based on blood test results using machine learning can have a significant impact on public health. This modern approach enables early disease detection, leading to shorter diagnostic times and faster treatment implementation.

Keywords: thyroid disease, classification, blood test, machine learning, artificial intelligence