BIMA
BULLETIN OF INTELLIGENT MACHINES AND ALGORITHMS
Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.30

Optimization Prediction of Differentiated Thyroid Cancer Using Machine Learning Algorithms and Hyperparameter Tuning

Rizal Habibulloh[1*], Budiman[2], Darsiti[3]

Informatics[1,2], Informatic Management[3]
Universitas Informatika dan Bisnis Indonesia, Bandung, Indonesia[1,2]
Universitas Teknologi Digital, Bandung, Indonesia[3]
https://unibi.ac.id/[1,2], https://digitechuniversity.ac.id/[3]
rizal.h22@student.unibi.ac.id[1*]

Abstract. Differentiated thyroid cancer is one of most common types of thyroid cancer among thyroid cancer survivors, with prevalence about 70 to 90%, indicating that DTC significantly dominates in terms of case numbers compared to other types of thyroid cancer. Therefore, a recurrence prediction system is needed to deliver high performance and possess high accuracy to provide accurate predictions despite the variations in each patient's medical data. This system must generate prediction class outputs that align with the actual condition so that survivors receive appropriate medical care. Thus, a web-based Clinical Decision Support System is one example of a system that can serve as a solution capable of handling medical decision-making based on machine learning methods, providing high accuracy and optimal performance in generating prediction classes that align with the patient's actual condition. In this study, the author selected six algorithms with a fairly diverse range; for example, the author's base models included KNN, support vector machines, logistic regression, and decision trees, while the ensemble learning model using the bagging technique included random forests and gradient boosting for boosting technique. Among the baseline models, logistic regression achieved an accuracy of 97.4%, making it the baseline model with the highest accuracy.

Key words: Thyroid Cancer, Machine Learning, Decision Support System, Clinical Decision Support System, Explainable AI.

1. INTRODUCTION

Thyroid cancer is a malignant tumor that grows from the cells of the thyroid parenchyma [1]. Its prevalence continues to rise wide globally, although the mortality rate has remained relatively stable in recent years. The clinical presentation of thyroid cancer varies widely, ranging from slow-growing, non-aggressive tumors to highly aggressive tumors with a relatively high mortality rate.

One of the most common types of thyroid cancer is differentiated thyroid cancer (DTC), also known as a type of cancer that develops from thyroid follicular cells—the cells that produce thyroid hormones. The term "differentiated" means that these cancer cells still look and behave like normal thyroid cells under a microscope and do not exhibit the typical signs of cancer cell growth. Between 70% and 90% of all thyroid cancer cases fall into this category, and it is characterized by relatively slow growth and a high cure rate [2].

A web-based prediction and classification system for differentiated thyroid cancer plays a crucial role in efforts to improve the early detection and management of thyroid cancer. This system is envisioned to serve as an innovative solution in the field of health technology, filling the gap between the high burden of thyroid cancer cases and the limited access to rapid, accurate, and affordable clinical diagnosis also known as Clinical Decision Support System (CDSS) [3].

Machine learning is broadly used to analyze complex data across disciplines [4]. In this study we will utilize this method to train model using classification method to determine whether a patient will have recurrence based on result training machine learning model in differentiated thyroid cancer recurrence dataset [5]. After we train all models then we implemented the best model to performed classification to generate prediction model on CDSS system. Therefore, this study aims to evaluate and compare the performance of several machine learning classification algorithms for predicting differentiated thyroid cancer recurrence and to investigate the effect of hyperparameter optimization on their predictive performance. The study also aims to implement the selected best-performing model

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License

Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.30

in a web-based CDSS as a prototype for supporting recurrence assessment.

The main contributions of this study are threefold. First, this study provides a comparative evaluation of six machine learning classification algorithms for differentiated thyroid cancer recurrence prediction using accuracy, precision, recall, and F1-score. Second, Randomized Search and Optuna are employed to investigate the effect of hyperparameter optimization on the performance of the evaluated models and to identify configurations that provide improved predictive performance. Third, the selected model is integrated into a web-based CDSS prototype to demonstrate its practical application for supporting differentiated thyroid cancer recurrence assessment.

2. RELATED WORK

Previous research plays a crucial role in this study; it serves as the conceptual framework or theoretical foundation to be developed in this research and can be used to compare current findings with those of previous studies in order to draw clear conclusions regarding novelty obtained in this study. Some previous studies that conducted similar research include Das et al., 2025, who achieved 96% accuracy using a decision tree model; Thakur et al., 2025, who achieved 98.2% accuracy using the same method; and Kumari et al., 2025, who achieved 93.6% accuracy using logistic regression. All three mentioned studies achieved very high evaluation metrics and demonstrated consistency in other machine learning evaluation metrics such as precision, recall, and F1 score[6] [7] [8]. However, all three study are still less outcome that might can be utilized by professional medical that treat differentiated thyroids cancer in order to predict recurrence in patient and give more detailed insight of some important variable that might have strong influence on patient recurrence. Hence this study aims to create a web-based application to predict the recurrence case accurately also giving strong analysist using Explainable AI to explain what factor to determine whether recurrence or no recurrence occurred on patient.

3. METHODS

As shown in Figure 1, the research followed a systematic workflow consisting of data collection, data preprocessing, machine learning model training, hyperparameter tuning, model evaluation, web-based application implementation, and application testing.

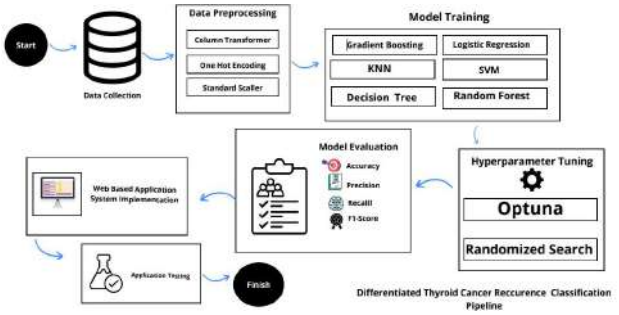


Fig. 1. Machine Learning Pipeline

3.1. Data Collection

This dataset consists of 383 patient records with 16 predictive features and one target variable [9]. The features represent demographic, lifestyle, clinical history, and staging information, while the target variable, Recurred, indicates whether differentiated thyroid cancer recurrence occurred. The distribution of the recurrence outcome is imbalanced, comprising 275 non-recurred cases (71.8%) and 108 recurred cases (28.2%). Therefore, stratified sampling was applied to preserve the proportion of the two outcome classes in the training and test sets. Detailed descriptions of the dataset features are presented in Table 1.

TABLE 1. Description of dataset used for differentiated thyroid cancer recurrence prediction.

Feature	Category	Type
Age	Demographic	Numerical
Gender	Demographic	Category
Smoking	Lifestyle	Category
Hx Smoking	Clinical History	Category
Hx Radiothreapy	Clinical History	Category
Thyroid Function	Clinical Finding	Category
Physical Examination	Clinical Finding	Category
Adenopathy	Clinical Finding	Category
Pathology	Pathological	Category
Focality	Pathological	Category
Risk	Stagging	Category
T	Stagging	Category
N	Stagging	Category
M	Stagging	Category
Stage	Stagging	Category
Response	Stagging	Category
Recurred	Target Feature	Category

3.2. Data Preprocessing

Pre-processing the data to make it more structured and cleaner so that it can be classified [10]. During this step, we will convert the data type of linguistic features into numerical form. The first step is data preprocessing, involving separating the features (X) from the target variable (Y), the Recurred column. Then, the target variable

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License

Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.30

undergoes a process called encoding, transforming the categorical Yes and No values into binary numerical values to understandable by machine learning model. The next step is to identify the numerical and categorical columns in the dataset.

To handle data type heterogeneity and prevent data leakage, we proceed by building a preprocessing stage using the Column Transformer module. In this stage, numerical features are scaled using StandardScaler, while categorical features are encoded using OneHotEncoder with the sparse_output parameter set to False so that the result is a dense array. Now that the preprocessor is defined, the data is split into a training set of 80% and a test set of 20% using stratification to ensure a proportional distribution of target classes in both sets, with random_state set to 42 for reproducibility. Finally, the preprocessor is applied to the training and test sets, and the transformed data is converted back into a Pandas DataFrame to preserve the feature name

3.3. Model Training

This stage involves selecting the machine learning algorithms to be used in the study, in this study we will use logistic regression, support vector machine, random forest, decision tree, k-nearest neighbors, and gradient boosting. This is because these six models have consistently produced high scores and are well-suited to this dataset, as demonstrated in previous related studies. All initial models before hyperparameter tuning are referred as baseline models. Before we train baseline models, the dataset was divided into training and testing subsets using a stratified 80:20 split. Accordingly, 80% of the observations were used for training, while the remaining 20% were reserved as an independent test set. Stratification was applied to maintain a similar distribution of the outcome classes in both subsets, and each model has configured parameter to enhanced model performances.

Each algorithms model has applied of some configurational called parameter which allow enhance model performance to classify more accurately and show correct prediction category as dataset which we will discuss in chapter 4.

Logistic regression has been used as a linear classification model in order to predict the probability of a class [11]. To ensure that the results of the experiment can be consistently reproduced, the parameter random_state=42 is used, allowing each training run to produce the identical model configuration during subsequent executions.

K-Nearest Neighbors classifies data based on its proximity to other samples [12]. This model employs the default parameter n_neighbors=5, meaning that classification decisions are determined based on the five nearest neighbors. The use of five neighbors aims to balance the model's sensitivity to noise and ability to generalize. A value of less than five may make the model overly sensitive to outliers, meanwhile a value of more than five may reduce the model's ability to capture local patterns in the data.

Decision Tree can be used to create classification rule sets in the form of a decision tree structure [13]. In this study, we implemented the parameter random_state=42 to enable consistent reproduction of the attribute selection and tree-building processes that involve random elements across all experiments.

Random Forest is an ensemble learning method that employs multiple decision trees to enhance classification performance [14]. The parameter random_state=42 is implemented to ensure consistency in the bootstrap sampling process and random feature selection for each tree, therefore allowing the experimental results to be replicated under identical conditions.

Gradient Boosting gradually builds a model by correcting the prediction errors of the earlier models [15]. The parameter random_state=42 helps to control the randomization process during model creation, allowing the training results to remain consistent and reproducible.

Support Vector Machines were used to find the optimal hyperplane that distinguished between data classes [16]. We implement the parameter random_state=42 to guarantee consistency in the experimental results, while probability=True is employed to enable the model to generate prediction probabilities. These probabilities are critical for analyses requiring estimates of the model's confidence level, and other probabilistic evaluations.

The dataset was divided into 80% training and 20% testing subsets using stratified sampling to preserve the proportion of the outcome classes in both subsets. The 5-fold Stratified K-Fold cross-validation was then applied exclusively to the 80% training set for model validation and model development. Stratification was used to maintain a consistent representation of both recurred and non-recurred cases in each fold. The cross-validation results were used to assess model performance across different training and validation folds and to identify potential signs of overfitting or underfitting. Throughout this process, the independent 20% test set was kept completely separate from both model training and cross-validation. The test set was used only for the final evaluation of the developed models on previously unseen data, thereby reducing the risk of data leakage and providing an unbiased assessment of model generalization.

3.4. Hyperparameter Tunning

In this stage, we implemented optimization techniques on the hyperparameters of the baseline model to improve accuracy using Random Search and Optuna. We defined a parameter search space (parameter distribution) for Random Search that consisted of a range of potential values for the key hyperparameters of each model [17], whereas Optuna is applied through the get_model function, which accepts a trial object in order to perform adaptive search based on Bayesian optimization [18]. Both optimization techniques were applied using the same training, validation, and testing configuration as the baseline models.

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License

Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.30

Specifically, the 80% training set was used for 5-fold Stratified K-Fold cross-validation, while the independent 20% test set was reserved exclusively for final evaluation. The hyperparameter settings were varied according to the respective search strategies to identify configurations that could improve model performance.

The first hyperparameter optimization method uses Randomized Search, conducted by defining a different parameter search space for each model. For the Logistic Regression and Support Vector Machine models, we use a uniform distribution to search for the C parameter, which controls regularization within ranges of 0.01 to 10 and 0.1 to 10, accordingly. For tree-based models such as Random Forest and Gradient Boosting, the search is performed on two parameters concurrently, namely n_estimators between 100 and 300 and max_depth between 3 and 15 for Random Forest, and between 3 and 10 for Gradient Boosting; both use the discrete randint distribution. Decision Tree optimizes only max_depth within the range of 3 to 20, while K-Nearest Neighbors optimizes n_neighbors within the range of 3 to 15. The entire search process configured with the n_iter parameter, where five parameter combinations were randomly selected; cv set to 5 for 5-fold cross-validation; the evaluation metric was accuracy; random_state set to 42 to guarantee reproducibility of results; and finally, n_jobs set to -1 to run the process in parallel, utilizing all available processor cores.

The second optimization is performed using the Optuna method using the get_model function, that accepts a trial object as an adaptive search mechanism based on Bayesian optimization. In the Logistic Regression model, the parameter C is optimized using suggest_float on a logarithmic scale ranging from 1e-3 to 10, and max_iter is set to 2000 to guarantee divergence. The support vector machine model optimizes two parameters concurrently, namely C within the same logarithmic range and the kernel, selectable categorically between linear and RBF. Random Forest optimizes n_estimators within the range of 100 to 300 and max_depth within the range of 3 to 20 using suggest_int. Decision Tree simply optimizes max_depth within the same range, while K-Nearest Neighbors optimizes n_neighbors within the range of 3 to 15. Gradient Boosting optimizes n_estimators within the range of 100 to 300 and learning_rate using suggest_float within the range of 0.01 to 0.3. The fundamental distinction between Optuna and Random Search lies in search mechanisms: Optuna adaptively narrows the search space based on previous trial results, enabling it to find optimal parameter combinations more efficiently than the purely random search employed by Random Search.

3.5. Model Evaluation

During this phase, assessment of the models will be conducted, focusing on the Accuracy, Precision, Recall, and F1-Score which is a standard for machine learning classification metrics [19]. Concurrently, the selection of the base model will also be performed at this stage to ensure the best model is chosen for implementation into the web-based application. This selection shall be based on the highest F1 Score, considering other factors such as low classification error, the three other metrics.

3.6. Web Based Application System Implementation

We implement selected base model in to web-based application using joblib library to create .pkl file after we run a python module which made the model accessible inside developed CDSS system so we can classify new data to generate prediction result more accurately.

3.7. Application Testing

At this final stage, testing was conducted to verify that the designed application functions as intended, as described in the previous section, and to make predictions based on the dataset, as well as to assess how well the model performs in making those predictions according to real data rows in dataset.

4. RESULTS AND DISCUSSIONS

In this Chapter, we will discuss the results of the research that has been conducted and present the base model selected based on the highest accuracy metric and consistent with the baseline model type to ensure that the model is indeed suitable for use in the CDSS application that has been developed.

4.1. RESULT

This section will present all four selected metrics to evaluate the results of each research model that has been implemented in baseline model and two optimization techniques, using the configurations described in the previous section. The metrics discussed in this section will be the same as those used in the baseline model, namely accuracy, precision, recall, and F1 score.

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License

Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.30

TABLE 2. Evaluation Matrix Result

Model	Methods	Accuracy	Precision	Recall	F1-Score
Logistic Regression	**Baseline**	**0.9740**	**0.974937**	**0.9740**	**0.9736**
	Randomized Search	0.9610	0.960884	0.9610	0.9607
	Optuna	0.9740	0.974937	0.9740	0.9736
K-Nearest Neighbors	Baseline	0.9480	0.948120	0.9480	0.9472
	Randomized Search	0.9480	0.948120	0.9480	0.9472
	Optuna	0.9480	0.948120	0.9480	0.9472
Decision Tree	Baseline	0.9090	0.910628	0.9090	0.9096
	Randomized Search	0.9090	0.910628	0.9090	0.9096
	Optuna	0.9220	0.923594	0.9220	0.9195
Random Forest	Baseline	0.9610	0.963054	0.9610	0.9601
	Randomized Search	0.9610	0.963054	0.9610	0.9601
	Optuna	0.9740	0.974937	0.9740	0.9736
Gradient Boosting	Baseline	0.9610	0.963054	0.9610	0.9601
	Randomized Search	0.9480	0.948120	0.9480	0.9472
	Optuna	0.9610	0.963054	0.9610	0.9601
Support Vector Machine	Baseline	0.9610	0.963054	0.9610	0.9601
	Randomized Search	0.9740	0.974937	0.9740	0.9736
	Optuna	0.9740	0.974937	0.9740	0.9736

Based on the evaluation results in Table 2, Logistic Regression with the baseline configuration was selected as the final model for the web-based application. Although Random Forest with Optuna and Support Vector Machine with Randomized Search and Optuna achieved the same accuracy of 97.4%, Logistic Regression obtained the same accuracy, precision, recall, and F1-score values without requiring hyperparameter optimization. Therefore, Logistic Regression was selected because it achieved comparable predictive performance using its baseline configuration, while maintaining a simpler model configuration for deployment. This result indicates that the baseline Logistic Regression model was already sufficiently effective for the given dataset. This indicates that the application of optimization techniques did not yield a significant improvement in the model's performance, suggesting that the initial configuration was already capable of delivering optimal performance. Overall, most models demonstrated good classification performance with metric values above 94%, where Logistic Regression, Support Vector Machine, and Random Forest achieved the best results, while Decision Tree produced relatively lower performance compared to the other models. First, we will test the base model to create following dataset rows to determine whether data could determine patient recurred or not recurred based on Table 3.

TABLE 3. Real DTC Patient Dataset sample

ID	T	N	Stage	Response	Recurred
P1	T1a	N1b	I	Structural Incomplete	Reccured
P2	T1a	N0	I	Excellent	No Reccured

Second, we input the data rows in P1 patient ID and the result is model could predicted accurately based on dataset on Table 3. Based on prediction in figure 2 we can conclude that the model has accurately predicting recurrence will be occurred and giving detailed judgement about contributor variable that led the prediction to Yes or Reccured.



Fig. 2. Result P1 Patient on CDSS Application

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License

Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.30

The last data rows we will input to our CDSS application is the P2 patient, and result is same as labelled from the dataset to predicting patient will no recurred and the model has given correct prediction result again to show No (No Recurred) which shown in figure 3.



Fig. 3. Result P2 Patient on CDSS Application

4.2. DISCUSSION

Table 4 will display the best model in all six algorithms we have chosen alongside with accuracy metric achieved in this study.

TABLE 4. Comparison Table

Algorithm	Study	Best Accuracy (%)	This Study (%)
Logistic Regression	This Study	97.40	97.40
Support Vector Machine	This Study	97.40	97.40
Random Forest	[7]	98.26	97.40
Decision Tree	[6]	96.00	92.21
K-Nearest Neighbors	This Study	94.81	94.81
Gradient Boosting	[7]	96.52	96.10

The results in Table 4 show that three of the six evaluated algorithms achieved improved accuracy after hyperparameter optimization. Decision Tree improved from 90.90% to 92.21%, while Random Forest increased from 96.10% to 97.40% and Support Vector Machine increased from 96.10% to 97.40%. These improvements indicate that hyperparameter tuning can be beneficial for models whose baseline configurations may not yet provide suitable decision boundaries for the dataset. In contrast, Logistic Regression, K-Nearest Neighbors, and Gradient Boosting did not show an improvement in accuracy after optimization. Logistic Regression achieved 97.40% with its baseline configuration, which was equal to the best accuracy obtained by Random Forest and Support Vector Machine after optimization. This suggests that the baseline Logistic Regression model was already well suited to the dataset and that further hyperparameter optimization did not provide additional predictive benefit. For K-Nearest Neighbors, the unchanged performance suggests that the tested range of neighborhood sizes did not substantially alter the classification results. Similarly, the absence of improvement in Gradient Boosting indicates that the tested combinations of the number of estimators and learning rate did not produce a measurable gain under the experimental setting. The Decision Tree showed the largest relative improvement among the models with lower baseline performance, although its optimized accuracy remained below that of Logistic Regression, Random Forest, and Support Vector Machine. A possible explanation is that the performance of a single decision tree can be sensitive to its maximum depth and may be more prone to overfitting than ensemble-based models. Overall, the results indicate that the effectiveness of hyperparameter optimization is model-dependent: some models benefited from parameter adjustment, whereas Logistic Regression already achieved competitive performance in its baseline configuration.

5. CONCLUSIONS

This study evaluates the performance of various machine learning classification algorithms for predicting differentiated thyroid cancer recurrence using multiple evaluation metrics. The results show that Logistic Regression achieved the highest baseline performance, with 97.40% accuracy, precision, recall, and F1-score, while Random Forest and Support Vector Machine achieved comparable performance after hyperparameter optimization. The findings demonstrate that hyperparameter optimization has a model-dependent effect, as some models improved after tuning while others did not show measurable improvement. This study has several limitations. First, the dataset contains a relatively limited number of patient records and an imbalanced distribution between recurred and non-recurred cases. Second, the evaluation was based on a single dataset and did not include external validation across different clinical settings. Third, although the study incorporated Explainable AI into the proposed application, further investigation is needed to assess the clinical interpretability and practical usefulness of the explanations with medical professionals. Future research should therefore investigate larger and more diverse clinical datasets, apply appropriate class-imbalance handling and advanced feature engineering techniques, and evaluate additional machine learning and deep learning approaches. External validation using data from different clinical institutions and more comprehensive assessment of model interpretability with medical professionals are also recommended to improve the clinical reliability and applicability of the proposed system.

REFERENCES

[1] Fatimah Eliana, Sabrina Azmi, Anggi Puspa Nur Hidayati, and Juhan Khalila, "Isolation, in Vitro Expansion, and Cryopreservation of Primary Cells Derived from Human Thyroid Carcinoma," *InaJEMD - Indones. J. Endocrinol. Metab. Diabetes*, vol. 2, no. 1, pp. 1–14, Jul. 2025, doi: 10.66266/inajemd.v2i1.35.

[2] R. C. Kusumadewi *et al.*, "Understanding Thyroid Carcinoma: Clinical Manifestations, Diagnosis, and Management," *J. Biol. Trop.*, vol. 24, no. 4, pp. 645–656, Oct. 2024, doi: 10.29303/jbt.v24i4.7664.

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License

Vol. 2 No.1 November 2026
E-ISSN: 3132-5115
DOI:10.65780/bima.v2i1.30

[3] Dieo Alfiky Ananda, Yani Maulita, and Suci Ramadani, "Sistem Pendukung Keputusan Pemilihan Klinik Kecantikan Menggunakan Metode WASPAS," *Bridge J. Publ. Sist. Inf. Dan Telekomun.*, vol. 2, no. 4, pp. 295–311, Sep. 2024, doi: 10.62951/bridge.v2i4.257.

[4] S. F. Pane, R. N. Zain, I. Setiawan, and sV. Putratama, "Predicting the Happiness Index Based on the HDI Indicator in Indonesia Using the Ensemble Learning Approach: Prediksi Indeks Kebahagiaan Berdasarkan Indikator IPM di Indonesia Menggunakan Pendekatan Ensemble Learning," *NUANSA Inform.*, vol. 19, no. 2, pp. 105–114, Jul. 2025, doi: 10.25134/ilkom.v19i2.410.

[5] S. Borzooei, G. Briganti, M. Golparian, J. R. Lechien, and A. Tarokhian, "Machine Learning for Risk Stratification of Thyroid Cancer Patients: a 15-year Cohort Study," *Eur. Arch. Otorhinolaryngol.*, vol. 281, no. 4, pp. 2095–2104, Apr. 2024, doi: 10.1007/s00405-023-08299-w.

[6] S. Das, A. K. Chaudhuri, N. R. Choudhury, and P. Ghosh, "Identification of the Recurrence of Differentiated Thyroid Cancer by Stacking Classifier," Jan. 03, 2025, *In Review*. doi: 10.21203/rs.3.rs-5713674/v1.

[7] D. Thakur, T. Gera, V. Bhardwaj, R. Mazen, A. Lasisi, and T. Engida, "A Comparative Study on Advanced Predictive Modeling of Thyroid Cancer Recurrence Using Multi Algorithmic Machine Learning Frameworks," *Sci. Rep.*, vol. 16, no. 1, p. 3385, Dec. 2025, doi: 10.1038/s41598-025-33396-7.

[8] H. Kumari, H. Kumari, and U. Nawarathne, "Differentiated Thyroid Cancer Recurrence Classification Using Machine Learning Models and Bayesian Neural Networks with Varying Priors: A SHAP-Based Interpretation of the Best Performing Model," 2025, *arXiv*. doi: 10.48550/ARXIV.2507.18987.

[9] S. Borzooei, G. Briganti, M. Golparian, J. R. Lechien, and A. Tarokhian, "Machine Learning for Risk Stratification of Thyroid Cancer Patients: A 15-Year Cohort Study," *Eur. Arch. Otorhinolaryngol.*, vol. 281, no. 4, pp. 2095–2104, Apr. 2024, doi: 10.1007/s00405-023-08299-w.

[10] E. Setiana, Marwondo, Venia Retreva Danestiara, and Wiyanudin, "Analisis Sentimen Pelaksanaan Kuliah Online Menggunakan Algoritma Support Vector Machine," *NUANSA Inform.*, vol. 17, no. 2, pp. 66–70, Jul. 2023, doi: 10.25134/ilkom.v17i2.11.

[11] F. Acito, "Logistic Regression," in *Predictive Analytics with KNIME*, Cham: Springer Nature Switzerland, 2023, pp. 125–167. doi: 10.1007/978-3-031-45630-5_7.

[12] R. K. Halder, M. N. Uddin, Md. A. Uddin, S. Aryal, and A. Khraisat, "Enhancing K-Nearest Neighbor Algorithm: A Comprehensive Review and Performance Analysis of Modifications," *J. Big Data*, vol. 11, no. 1, p. 113, Aug. 2024, doi: 10.1186/s40537-024-00973-y.

[13] E. Halabaku and E. Bytyçi, "Overfitting in Machine Learning: A Comparative Analysis of Decision Trees and Random Forests," *Intell. Autom. Soft Comput.*, vol. 39, no. 6, pp. 987–1006, 2024, doi: 10.32604/iasc.2024.059429.

[14] T. Becker, A.-J. Rousseau, M. Geubbelmans, T. Burzykowski, and D. Valkenborg, "Decision Trees and Random Forests," *Am. J. Orthod. Dentofacial Orthop.*, vol. 164, no. 6, pp. 894–897, Dec. 2023, doi: 10.1016/j.ajodo.2023.09.011.

[15] R. Sibindi, R. W. Mwangi, and A. G. Waititu, "A Boosting Ensemble Learning Based Hybrid Light Gradient Boosting Machine and Extreme Gradient Boosting Model for Predicting House Prices," *Eng. Rep.*, vol. 5, no. 4, p. e12599, Apr. 2023, doi: 10.1002/eng2.12599.

[16] O. A. Montesinos López, A. Montesinos López, and J. Crossa, "Support Vector Machines and Support Vector Regression," in *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, Cham: Springer International Publishing, 2022, pp. 337–378. doi: 10.1007/978-3-030-89010-0_9.

[17] M. S. Anggreainy, "Hyperparameter Optimization for Stress Level Classification in University Students: A Comparative Analysis of GridSearchCV and RandomizedSearchCV," in *2025 Tenth International Conference on Informatics and Computing (ICIC)*, Lampung, Indonesia: IEEE, Oct. 2025, pp. 1–6. doi: 10.1109/ICIC68054.2025.11309700.

[18] R. De A. Fernandes, F. R. Seppe, C. T. C. Júnior, I. G. Torné, C. D. S. Filho, and S. B. T. Rego, "A Bayesian Optimization Approach of Ensemble and Decision Tree Learning Applied to Industrial Energy Consumption Prediction," in *2023 15th IEEE International Conference on Industry Applications (INDUSCON)*, São Bernardo do Campo, Brazil: IEEE, Nov. 2023, pp. 842–849. doi: 10.1109/INDUSCON58041.2023.10374661.

[19] S. Sudriyanto, A. Fatah, and M. D. W. Putra, "Evaluasi Model Machine learning untuk Prediksi Keparahan Kanker Berdasarkan Data Real-world Global," *J. Kridatama Sains Dan Teknol.*, vol. 7, no. 02, pp. 922–935, Dec. 2025, doi: 10.53863/kst.v7i02.1940.

Published by Maheswari Publisher
Creation disseminated under Creative Commons Attribution 4.0 International License