

Manuscript received May 2, 2024; revised May 23, 2024; accepted May 27, 2024; date of publication October 2, 2024
Digital Object Identifier (DOI): <https://doi.org/10.35882/jeeemi.v6i4.467>

Copyright © 2024 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License ([CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)).

How to cite: Nor Aida, Triando Hamonangan Saragih, Dwi Kartini, Radityo Adi Nugroho, Dodon Turianto Nugrahadi, "Comparison of Extreme Machine Learning and Hidden Markov Model Algorithm in Predicting The Recurrence Of Differentiated Thyroid Cancer Using SMOTE", Journal of Electronics, Electromedical Engineering, and Medical Informatics, vol. 6, no. 4, pp. 429-444, October 2024.

Comparison of Extreme Machine Learning and Hidden Markov Model Algorithm in Predicting The Recurrence Of Differentiated Thyroid Cancer Using SMOTE

Nor Aida^{ORCID}, Triando Hamonangan Saragih^{ORCID}, Dwi Kartini^{ORCID}, Radityo Adi Nugroho^{ORCID}, Dodon Turianto Nugrahadi^{ORCID}.

Computer Science Department, Lambung Mangkurat University, Banjarbaru, South Kalimantan, Indonesia

Corresponding author: Triando Hamonangan Saragih (e-mail: triando.saragih@ulm.ac.id).

Lambung Mangkurat University supported this work by providing resources and support.

ABSTRACT Differentiated thyroid cancer is the most common type of thyroid cancer; the types in this category are papillary, follicular, and hurthel cell carcinoma. Up to 20% of DTCs will experience recurrence, although this figure reduces to 5% in low-risk patients. There is still little research on thyroid cancer prediction using a machine learning approach, especially the prediction recurrence of DTCs. This research aims to compare the performance of the Extreme Learning Machine and the Hidden Markov Model using SMOTE in predicting the recurrence of DTCs. The dataset used in this research is differentiated thyroid cancer recurrence from Kaggle. This research methodology comprises preprocessing, data sharing, SMOTE, ELM and HMM modeling algorithms, and evaluation. ELM with SMOTE gets the best results at a ratio of 90:10 with 35 hidden neurons that get an accuracy value of 1.00, precision 1.00, recall 1.00, and AUC 1.00. ELM modeling gets the best results at a ratio of 90:10 with 45 hidden neurons that get an accuracy value of 1.00, precision 1.00, recall 1.00, and AUC 1.00. HMM modeling with SMOTE gets the best results at a ratio of 70:30 with two hidden states and two iterations, with an accuracy value of 0.8696, precision of 0.8832, recall of 0.7848, and AUC of 0.9174. Last, HMM modeling gets the best value at a ratio of 60:40 with two hidden states and three iterations, which get an accuracy value of 0.8636, precision 0.8471, recall 0.7946, and AUC 0.9343. Based on the results of this study, it can be concluded that ELM with SMOTE gets the best performance, followed by ELM without SMOTE, HMM with SMOTE, and HMM without SMOTE. The implication is that ELM with SMOTE can produce high accuracy in predicting the recurrence of DTCs.

INDEX TERMS Thyroid cancer, Extreme Learning Machine, Hidden Markov Model, SMOTE

I. INTRODUCTION

The thyroid gland is a small butterfly-shaped gland located at the base of the neck, just below the Adam's apple. It is part of the endocrine system, responsible for producing hormones. Thyroid hormones regulate metabolism, growth and development, and brain and heart function. When cells in the thyroid gland grow abnormally, it can lead to thyroid cancer. In 2022, there were approximately 821.173 new cases of thyroid cancer diagnosed globally, accounting for approximately 4% of all cancer cases. thyroid cancer ranks as the seventh most common cancer in terms of incidence overall and fifth in women [1]. About 75% of patients diagnosed with

thyroid cancer are women, with an incidence rate of 10.1/100,000 women per year, while the incidence rate in men is 3.1/100,000 per year [2]. Among the different types of thyroid cancer, differentiated thyroid cancer (DTCs) is the most common. The types of thyroid cancer that fall under this category are papillary, follicular, and hurthel cell carcinoma [3].

Although the cause of thyroid cancer is unknown, many studies have evaluated its risk factors. High-risk factors for DTC are radiation exposure of the head and neck region, chromosomal alterations such as RAS and BRAF gene mutations and PAX8/PPAR γ fusion protein expression, and

hereditary conditions such as medullary thyroid cancer, syndromic and non-syndromic familial non-medullary thyroid cancer [4]. Traditionally, prophylactic cervical lymph node dissection, adjuvant radio-iodine ablation, and thyroid-stimulating hormone (TSH) suppression with levothyroxine have been suggested as treatment methods to reduce the recurrence rate of DTC [5].

The prognosis of DTCs is generally good, and the mortality rate, although slightly increased in recent years, is generally low. For this reason, attention is mainly focused on the risk of recurrence, which the American Thyroid Association (ATA) risk classification predicts. Specifically, up to 20% of DTCs will experience recurrence, although this figure reduces to 5% in low-risk ATA patients [6]. One practical approach to analyzing and addressing such health cases is machine learning.

Comprehensive research was conducted using machine learning in cancer detection. Research on thyroid cancer detection has been conducted by [7] with the highest accuracy score of 87% and AUC of 0.93 using the Logit Boost algorithm. Other research on thyroid cancer was conducted by [8] using the CNN algorithm with an accuracy score of 99%. Research on differentiated thyroid cancer recurrence was conducted by [9] using several algorithms, including Logistic Regression with 91% accuracy, Naive Bayes with 86% accuracy, Decision Tree with 91% accuracy, and KNN with 90% accuracy. There is still little research on thyroid cancer prediction using a machine learning approach, especially the prediction of thyroid cancer recurrence for differentiated thyroid cancer patients using the Extreme Learning Machine algorithm and Hidden Markov Model algorithm.

A method that is often used in machine learning is a Neural Network, often referred to as Artificial Neural Network (ANN). In an ANN, nodes are usually organized into layers, with each layer receiving input from the previous layer and passing its output to the next layer. Extreme Machine Learning (ELM) and Hidden Markov Model (HMM) are two algorithms that use the concept of ANN, both based on the concept of a network of nodes, where each node represents a mathematical function that receives some input and produces an output.

Extreme Machine Learning (ELM) has a much faster training phase than other more common ANN algorithms, where, instead of using a long gradient-based approach like the back-propagation algorithm, ELM uses the pseudo inverse of the hidden layer to analytically determine the output weights [10]. ELM uses the concept that the input weights and bias values in the hidden layer are determined randomly, while the output weight values are calculated by utilizing the pseudo inverse of Moore-Penrose (MP) [11].

Research conducted by [12] to predict breast cancer using the ELM with the Wisconsin breast cancer diagnosis dataset, this study obtained an accuracy value of 98% on hidden neurons 250 and an F1 score of 0.81. Another study was conducted by [13] to classify epilepsy through physical

activity using epilepsy using the ELM with an accuracy score of 100% at 920 hidden neurons and an AUC score of 0.99.

Hidden Markov Model (HMM) is a statistical model designed using the Markov process with hidden states. Markov models adapt the concept of less-memory properties, where the transition from one state to another depends only on the current state. In HMM, the emitted symbols are observable, and the random transition from one state to another remains unobservable [14]. HMMs provide researchers with a probabilistic framework for inference and prediction and require fewer training samples, and thus have been applied to behavior recognition, speech recognition, and other fields [15].

Research on the HMM was conducted by [16] to classify Koranic recitation phenomena with a dataset of Koranic phonemes containing 21 verses from 30 reciters; this study achieved an accuracy value of 100% and a precision value of 100. Another study was conducted by [17] to predict traffic congestion on neighboring roads using the HMM, the results showed the accuracy value reached 89%.

In the medical and healthcare domains, the rarity of certain conditions or diseases naturally results in fewer instances of positive cases compared to normal or negative cases, leading to an imbalanced dataset. Such imbalances present unique challenges in machine and deep learning, as standard algorithms optimized for balanced datasets may not perform effectively, often overlooking the minority class, which usually represents the most crucial information to be predicted [18]. The dataset used is the Differentiated Thyroid Cancer Recurrence dataset taken from Kaggle. The dataset has imbalanced data in one of its classes, so a technique is needed to handle the problem. One of the techniques used to overcome class imbalance in data is the creation of synthetic data from the minority class, which is SMOTE. It works by augmenting the number of data instances in the minority class through the generation of synthetic data points from its nearest neighbors using Euclidean distance [19]. In research on SMOTE conducted by [20] to predict MBTI personality, this study compared several classification algorithms with SMOTE and without SMOTE. The Logistic Regression algorithm achieved the highest accuracy value of 82.82%, while Logistic Regression with SMOTE achieved an accuracy value of 83.37%.

Based on the above, this research will use an Extreme Learning Machine and Hidden Markov Model with the SMOTE oversampling method to improve the performance of DTC recurrence prediction. The objective of this study is to compare the performance of ELM and HMM methods, both without and with the application of SMOTE in predicting the recurrence of differentiated thyroid cancer. The specific objectives in this study are to identify the best performing method and overcome the data imbalance often encountered in medical datasets increase the early detection rate of recurrence and improve the quality of life of patients.

Research question: Which of the ELM and HMM methods is best for predicting the recurrence of differentiated thyroid

cancer? What is the effect of applying SMOTE to ELM and HMM in predicting this recurrence? The contributions of this research are:

1. Providing knowledge about the effect of ELM and HMM algorithms on the prediction of the recurrence of differentiated thyroid cancer.
2. Providing the knowledge about the effect of SMOTE on the prediction of the recurrence of differentiated thyroid cancer.
3. Providing knowledge about the relationship between data science modeling and the health field, potentially improving patient care and health outcomes.

II. METHOD

The flowchart of this research can be represented in **FIGURE 1**. This research employs Python for machine learning classification on the "Differentiated Thyroid Cancer Recurrence" dataset from Kaggle. The methodology comprises several stages. Initially, data preprocessing is conducted to convert categorical values into a numerical format using encoding techniques. Subsequently, the dataset is divided into training and testing sets across various proportions (90:10, 80:20, 70:30, 60:40, and 50:50) for comparative analysis. Due to class imbalance, SMOTE is applied to balance the data, and its impact on model performance is evaluated. Extreme Learning Machine and Hidden Markov Model are utilized for classification, and their outcomes are assessed using accuracy, precision, recall, and AUC metrics on the testing data.

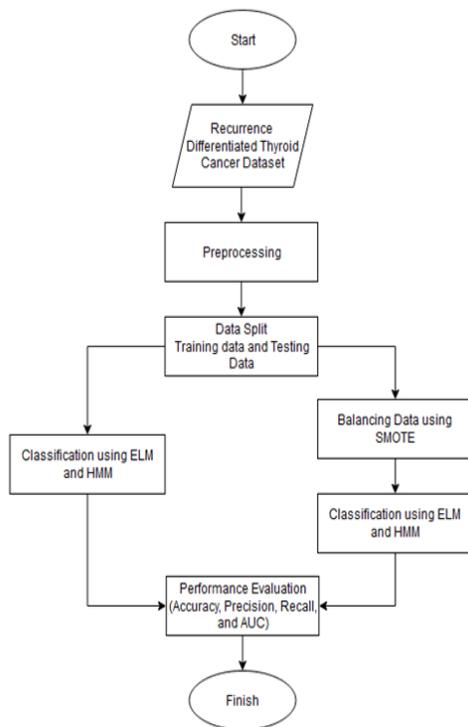


FIGURE 1. Research Flowchart

A. DATA COLLECTION

The data used in this study is the Differentiated Thyroid Cancer Recurrence dataset taken from the Kaggle website by Shiva Borzooei, Giovanni Briganti, Mitra Golparian, Jerome R. Lechien, and Aidin Tarokhian published in 2023. This dataset can be accessed through the following link: <https://www.kaggle.com/datasets/joebeachcapital/differentiated-thyroid-cancer-recurrence>. The data was collected over 15 years, and each patient was monitored for at least 10 years. The data consists of 383 records with 16 features and the last 1 class is the target class, which is the recurrence status of differentiated thyroid cancer. The number of majority classes in this dataset is 275 (no), while the number of data in the minority class is 108 (yes). The percentage ratio for the (yes) and (no) classes is 72% and 28%, indicating an imbalance of classes in the data. The graph of data comparison on the Recurrence feature can be seen in **FIGURE 2**.

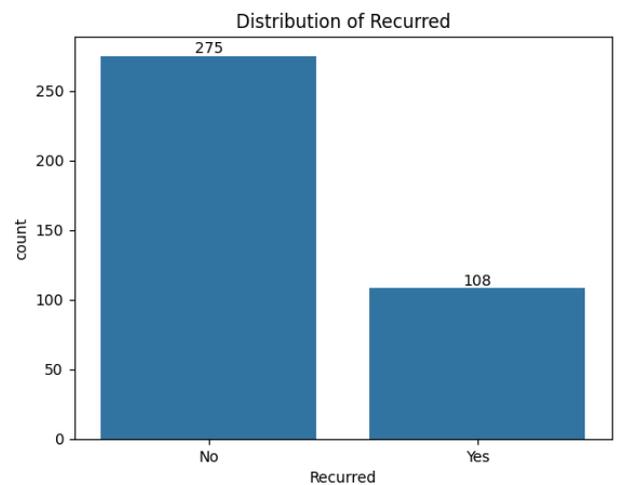


FIGURE 2. Distribution of Recurred Feature

This dataset is already good because there are no missing values, split data, or duplicate data, so the preprocessing carried out only transforms category data into numeric data with the encoding method. Details of the dataset features can be found in **TABLE 1**.

TABLE 1
Detail Dataset of Differentiated Thyroid Cancer Recurrence

Variable	Feature Name	Range (count)
	Age	15 - 82
	Gender	Female (312), Male (71)
	Smoking	Yes (49), No (334)
	Hx Smoking	Yes (28), No (355)
Input	Hx Radiotherapy	Yes (7), No (376)
	Thyroid Function	Clinical Hyperthyroidism (20), Clinical Hypothyroidism (12), Euthyroid (332), Subclinical Hyperthyroidism (5), Subclinical Hypothyroidism (14)
	Physical Examination	Diffuse goiter (7), Multinodular goiter (140), Normal (7), Single nodular goiter-left (89), Single nodular goiter-right (140)

TABLE 1
 (Continued)

Variable	Feature Name	Range (count)
Input	Adenopathy	Bilateral (32), Extensive (7), Left (17), No (277), Posterior (2), Right (48)
	Pathology	Follicular (28), Hurthel cell (20), Micropapillary (48), Papillary (287)
	Focality	Uni-Focal (247), Multi-Focal (136)
	Risk	High (32), Intermediate (102), Low (249)
	T	T1a (49), T1b (43), T2 (151), T3a (96), T3b (16), T4a (20), T4b (8)
	N	N0 (268), N1a (22), N1b (93)
	M	M0 (365), M1 (18)
	Stage	I (333), II (32), III (4), IVA (3), IVB (11)
	Response	Biochemical Incomplete (23), Excellent (208), Indeterminate (61), Structural Incomplete (91)
Output	Recurred	Yes (108), No (275)

B. PREPROCESSING

Data preprocessing aims to optimize the data used by various ML algorithms and ultimately improve accuracy with lower

computational performance [21]. The data used in this study is quite good because there are no missing values or duplicate data. So, the preprocessing done in this research transforms category variables into numeric data using the encoding method. The preprocessing step in this study uses the library from sklearn.preprocessing from scikit-learn. For features that have two classes, such as “yes” and “no” use label encoding, while for features that have more than two classes use a one-hot encoding. Label encoding simply assigns an integer value to each possible value of a categorical variable [22]. One-hot encoding, on the other hand, creates a new variable for each categorical feature level, and each category is mapped to a binary variable containing 0 or 1 [23].

The label encoding method is applied to the features Age, Gender, Smoking, Hx Smoking, Hx Radiotherapy, Focality, and Recurred. While the one-hot encoding method is applied to the features of Thyroid Function, Physical Examination, Adenopathy, Pathology, Risk, T, N, M, Stage, and Response. So the dataset, which initially had 16 features and one target class, turned into 50 features and one target class. TABLE 2 shows the sample dataset "Differentiated Thyroid Cancer Recurrence" before preprocessing, and TABLE 3 shows the sample dataset "Differentiated Thyroid Cancer Recurrence" after preprocessing.

TABLE 2
 Dataset Sample Before Preprocessing

Age	Gender	Smoking	Hx Smoking	Thyroid Function	Physical Examination	...	Pathology	Focality	Risk	T	Recurred
27	F	No	No	Euthyroid	Single nodular goiter-left	...	Micropapillary	Uni-Focal	Low	T1a	No
34	F	No	Yes	Euthyroid	Multinodular goiter	...	Micropapillary	Uni-Focal	Low	T1a	No
62	F	No	No	Euthyroid	Single nodular goiter-right	...	Micropapillary	Uni-Focal	Low	T1a	No
52	M	Yes	No	Euthyroid	Multinodular goiter	...	Micropapillary	Multi-Focal	Low	T1a	No
...
81	M	Yes	No	Euthyroid	Multinodular goiter	...	Papillary	Multi-Focal	High	T4b	Yes
72	M	Yes	Yes	Euthyroid	Multinodular goiter	...	Papillary	Multi-Focal	High	T4b	Yes
61	M	Yes	Yes	Clinical Hyperthyroidism	Multinodular goiter	...	Hurthel cell	Multi-Focal	High	T4b	Yes
67	M	Yes	No	Euthyroid	Multinodular goiter	...	Papillary	Multi-Focal	High	T4b	Yes

TABLE 3
 Dataset Sample After Preprocessing

Age	Gender	Smoking	Hx Smoking	Thyroid Function_Euthyroid	Physical Examination_Single nodular goiter-right	...	Adenopathy_Bilateral	Pathology_Papillary	Risk_High	T_T4b	Recurred
27	0	0	0	1	0	...	0	0	0	0	0
34	0	0	1	1	0	...	0	0	0	0	0
62	0	0	0	1	1	...	0	0	0	0	0
52	1	1	0	1	0	...	0	0	0	0	0
81	1	1	0	1	0	...	0	1	1	0	1
72	1	1	1	1	0	...	1	1	1	0	1
61	1	1	1	0	0	...	0	0	1	0	1
67	1	1	0	1	0	...	1	1	1	0	1

C. DATA SHARING

Before performing classification, the dataset is divided into two parts, namely training data and testing data. Training data is used to train the machine learning algorithm model while testing data is used to evaluate the performance of the pre-trained model. Using split data, training data, and testing data in this study are divided into several proportions, which are 90:10, 80:20, 70:30, 60:40, and 50:50. The data sharing process in this study uses the `train_test_split` library from `skicit-learn`.

D. SMOTE

Class imbalance is defined as a skewed distribution of instances found in a data set among classes in binary and multiclass problems. This asymmetry in class distribution negatively impacts classifier performance, especially in multiclass problems. The problem would come from the fact during the learning phase, the classifier is optimized to maximize the objective function, with overall accuracy being the most common. Unbalanced learning has been addressed in three different ways: over/undersampling, cost-sensitive training, and changes/adaptations in the learning algorithm. Since resampling strategies represent a set of methods that are independent of the classifier by operating at the data level, they allow the use of any available algorithm without requiring any type of change or adaptation to the algorithm. In particular, in the case of oversampling, the user can balance the class distribution of the data set without losing information, which is not the case with undersampling techniques [24].

Synthetic Minority Oversampling Technique (SMOTE) is an approach that uses 'synthetic' examples to oversample minority classes to resolve imbalanced data. Using synthetic examples in 'feature space' rather than 'data space' means that SMOTE is performed based on the values and characteristics of the data relationships rather than focusing on all data points. SMOTE works by inserting synthetic cases along the line connecting any or all of the k -nearest neighbors of each minority class and over-sampling each minority class. The k -nearest neighbors are randomly selected based on the amount of oversampling required [20].

The SMOTE algorithm was proposed by Chaw La in 2002, and it is an improvement method based on ROS (Random Over Sampling). In the SMOTE algorithm, new samples are generated based on the original samples, which has a greater probability of obtaining effective features than random new sampling. The SMOTE algorithm selects a line connecting the two original samples as the new sample range and determines a point on the line as the new sample. The steps of the SMOTE algorithm are as follows [25]:

Step 1: For each sample x in the training set, calculate the Euclidean distance to each minority class sample x_i , and get the k nearest neighbors of each minority class sample.

Step 2: According to the degree of sample imbalance, set the sampling ratio N . For x_i , randomly select N samples from its k nearest neighbors, denoted by x_h .

Step 3: Based on Eq. (1) [25], generate new samples based on x_i and x_h until the classes are balanced, denoted by x_{new} .

$$x_{new} = x_i + rand(0,1) \times (x_h - x_i) \quad (1)$$

where x_{new} is New synthetic sample, x_i is Original minority sample, and x_h is Nearest neighbour sample.

SMOTE generates fictional data based on the space characteristic similarities of minority modules [26]. Over the past decade, SMOTE has proven its utility across multiple domains, resulting in significant contributions to a wide range of applications [27]. This method helps to balance out the class distribution, thus allowing classifiers to learn more effectively from the data and improving their ability to generalize to minority class instances [28].

In the dataset used in this study, there is imbalanced data on the 'Recurred' feature with a ratio of 72% for the 'No' class and 28% for the 'Yes' class, applying the SMOTE technique to balance the data is necessary. After several experiments using the accuracy, precision, recall, and AUC values to determine how many k -nearest neighbors and random states are the most optimal, the k -nearest neighbor is 5 and the random state is 42 for modeling with Extreme Learning Machine, while the Hidden Markov Model modeling obtained the most optimal k -nearest neighbor is 4 and the random state is 42. For research with the sharing of training data and testing data with a proportion of 90:10, the 'Yes' and 'No' classes each have 247 data; at a proportion of 80:20, each class has 217 data; at a proportion of 70:30, each class has 192 data; at a proportion of 60:40, each class has 163 data; and at a proportion of 50:50, each class has 133 data.

E. CLASSIFICATION

In machine learning, classification refers to the prediction problem of determining the class to which samples from a data set will be assigned. A classifier algorithm must be provided with training data with labeled classes. Then, the classifier can predict the class for new test data based on the training data. This approach is called supervised learning, and classification is one example of such a method. The training set is chosen as a subset of the overall data set. The general approach is to divide the known samples into training and testing sets, following some general principles about the ratio of the two. Ultimately, the test set includes a much smaller number of samples than the training set (preferably, the test set and training set should be separate). The test set is used to evaluate the classification quality. Classification algorithms for prediction problems are evaluated based on performance. Various measures to assess classification quality can be used depending on the situation. One of the most commonly used measures is accuracy, which determines how many samples from the entire data set are correctly classified into the appropriate class [29].

This research uses Extreme Learning Machine and Hidden Markov Model classification algorithms to predict the recurrence of differentiated thyroid cancer. Both ELM and HMM use the basic concept of a network consisting of nodes (neurons). Each node receives input, involves a mathematical

pre-calculation process, and produces an output. The reason for choosing these two classification algorithms for predicting the recurrence of differentiated thyroid cancer is because the mathematical model of ELM is simpler and more effective in predicting the output quickly and accurately. In addition, although HMM is highly complex, it can predict unobserved states, such as factors that affect the state of cancer. Several algorithms can be used to predict thyroid cancer recurrence, such as Neural Network, Random Forest, or Support Vector Machine. However, ELM and HMM have better advantages in predicting thyroid cancer recurrence. ELM can predict the output quickly and accurately, while HMM can predict the unobserved state directly.

1. EXTREME LEARNING MACHINE

Recently, there has been an approach called Extreme Machine Learning (ELM) for training Single-Layer Feedforward Neural Network (SLFN). ELM introduces a unique approach where hidden nodes are initialized randomly and kept fixed without repeated tuning [30]. ELMs have become popular because they deliver relatively high accuracy and cross-domain adaptation with low time consumption. In addition, ELMs have superior generalization capabilities and less training time than deep network models, as they do not require iterative processes to tune parameters [31].

ELM requires a sufficient number of neurons in the hidden layer to obtain good performance and fast convergence. The difference between ELM and other traditional Machine Learning (ML) models that typically use gradient descent-based algorithms is that ELM uses randomly determined input weights and bias values that do not change during the learning process. This approach prevents some of the problems that usually accompany gradient descent methods, such as repeated adjustments to the weight and bias values, staying at a local minimum, and slowing down the speed of convergence. However, the correct number of neurons to create the hidden layer is still one of the open questions facing ELMs [11]. The architecture diagram of the Extreme Machine Learning algorithm can be seen in FIGURE 3 below [32]:

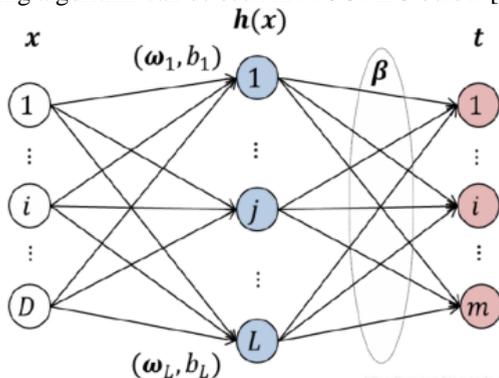


FIGURE 3. The Architecture Diagram of Extreme Learning Machine

The stages of the ELM algorithm are generally as follows [33]:

- Given a training set $N = \{(x_i, t_i) \mid x_i \in R^n, t_i \in R^m, i = 1, \dots, N\}$, an activation function, and the number of hidden layers N .
1. Set the input weights (w_i) and bias ($b_i, i = 1, \dots, N$).

2. Randomly assign input weight values in the value range -1 to 1 and bias values in the value range 0 to 1.
2. Calculate the output matrix of the hidden layer. Calculate the result of the output matrix of the hidden layer (H_{init}) using Eq. (2) [33].

$$H_{init} = X \times W^T + b \tag{2}$$

H_{init} = Hidden layer output matrix
 X = Input data
 W^T = Transpose of weight matrix
 b = bias

Then, the resultant hidden layer output matrix is activated with a specific activation function. The purpose of the activation function is to incorporate nonlinearity into the model. This research uses a binary sigmoid activation function, the formula of the binary sigmoid activation function is written in Eq. (3) [33].

$$H = \frac{1}{1 + \exp^{-H_{init}}} \tag{3}$$

H = Activated hidden layer output matrix

3. Calculating the output weights (β). The output layer weights written in Eq. (5) [33] are calculated using the generalized Moore-Penrose inverse of the hidden layer output matrix written in Eq. (4) [33].

$$H^+ = (H^T H)^{-1} H^T \tag{4}$$

H^+ = Generalised Moore-Penrose inverse of H matrix
 H^T = Transpose of H matrix
 H = Activated hidden layer output matrix

$$\beta = H^+ T \tag{5}$$

β = Output weight matrix
 T = Target value

After training, the next step is testing data, which aims to evaluate the previous ELM method training process. Given a set of testing data $N = \{(x_i, t_i) \mid x_i \in R^n, t_i \in R^m, i = 1, \dots, N\}$. The steps of the testing process are as follows:

1. Initialize the input weights (w_i) and bias (b) from the previous training process.
2. Calculate the output matrix of the hidden layer.
3. Calculating the predicted $f(x)$ written in Eq. (6) [33] value that will be compared with the target value, the output weight value used is obtained from the previous training process.

$$f(x) = H\beta \tag{6}$$

$f(x)$ = Predicted value
 H = Activated hidden layer output matrix
 β = Output weight matrix

The strength of ELM is that the absence of iteration makes the training process simpler enabling faster and more efficient prediction of the recurrence of differentiated thyroid cancer faster and with high efficiency. The weakness of ELM is that it requires more hidden neurons to achieve the best performance than conventional neural networks [34].

ELM modeling in this study uses the ELMClassifier library from Skicit-ELM, which tests the number of hidden neurons from 5 to 50 with a multiple of 5. To ensure that the research results obtained are optimal, several parameter adjustment experiments were carried out so that the activation function used was Binary Sigmoid, the random state was 42 for ELM modeling without SMOTE, and the random state was 43 for ELM modeling with SMOTE.

2. HIDDEN MARKOV MODEL

HMM is a mathematical model that infers the pattern of the observed state sequence by assuming that an unknown state sequence can produce a known observation sequence. HMMs are based on Markov processes consisting of multiple states and the transition relationships between them. Given this, HMMs provide researchers with a probabilistic framework for inference and prediction and require fewer training samples, and thus have been applied to behavior recognition, speech recognition, natural language processing, error detection, economic forecasting, earthquake prediction, hydrological prediction, and other fields [15].

In an HMM, the system is represented as a Markov process with states invisible to the observer but with visible outputs (observations) that are random functions of the states. HMM can be defined as a machine learning model, specifically as a discrete method, which is a statistical strategy for modeling systems intended to contain Markov processes with hidden states [35]. In addition, in the application of HMMs, ANNs, and genetic algorithms are combined to forecast financial market behavior, which can be applied to in-depth stock market analysis. HMMs are also used for credit card fraud detection, where HMMs are trained with 'normal' cardholder behavior to identify suspicious operations detected by the low probability given by the HMM [36].

The five essential components that make up the hidden Markov model are the number of states, the number of distinct observations, the state transition model, the observation model, and the initial state distribution. To determine the observation probabilities, the forward algorithm is adopted, while to predict the sequence of hidden states in the available data, the Viterbi algorithm is used. The learning stage of the HMM is performed using the EM algorithm [37].

HMM is a statistical model that describes a Markov process with unknown parameters. The model contains two random lines: an unobservable hidden line I and an observable line O. The hidden line I is a random line that can affect the observable line O and cannot be directly observed or obtained. The observable row O is a random row that can be directly observed or obtained under the influence of the hidden row I. In HMM, the form of the hidden row I is written in Eq. (7) [38]:

$$I = (i_1, i_2, \dots, i_T) \quad (7)$$

The shape of the O line that can be observed in Eq. (8) [38]:

$$O = (o_1, o_2, \dots, o_T) \quad (8)$$

In HMM, the hidden Markov chain randomly generates the hidden lineup I, and then the observable lineup O is generated

from each state of the hidden lineup. FIGURE 4 shows the relationship between the hidden lineup and the observable lineup in HMM.

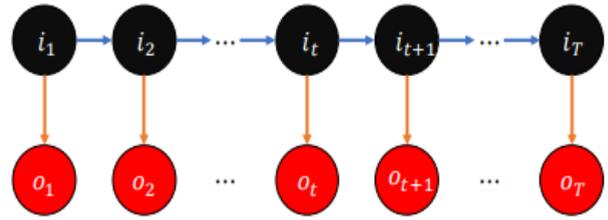


FIGURE 4. The Relationship Between The Hidden Line and The Observable Line in An HMM

The HMM can be expressed by an initial probability distribution π , a state transition matrix A, and an emission matrix B. Where the initial probability distribution π written in Eq. (9) [38] represents the probability distribution of the initial hidden state:

$$\pi = (p_1, p_2, \dots, p_N) \quad (9)$$

The state transition matrix A written in Eq. (10) [38] represents the probability that the hidden sequence is q_j at time $t + 1$ in the case that the hidden state is q_i at time t :

$$A = [a_{ij}], a_{ij} = P(i_{t+1} = q_j | i_t = q_i) \quad (10)$$

The emission matrix B written in Eq. (11) [38] represents the probability that the observation sequence is v_k when the hidden state is q_j at time t :

$$B = [b_i(k)], b_i(k) = P(o_t = v_k | i_t = q_i) \quad (11)$$

Considering all of the above, for convenience, the following compact notation is typically used to denote the entire set of parameters that characterize the HMM written in Eq. (12) [39]:

$$\lambda = (A, B, \pi) \quad (12)$$

Depending on how the HMM is applied, there are three different problems that can be defined [40]:

1. Evaluation problem
The goal is to compute the likelihood $p(O|\lambda)$ for a given set of observations O through. In general, a forward-backward algorithm is applied to solve this problem.
2. The learning problem
Involves adjusting the inputs of the modes λ to improve the likelihood of the observation series $p(O|\lambda)$. Expectation-maximization (EM) algorithms are often used to learn HMMs.
3. Decoding problem
What is the most probable state set of the model λ that generates the observation series O? The Viterbi algorithm is mainly applied to solve this problem. Viterbi algorithm can generate the hidden state sequence that best matches the observed state sequence, identifying the most probable path corresponding to the observed trajectory in the road network [41].

Hidden Markov Models (HMMs) are highly effective in predicting unobserved states, a crucial function in various applications such as disease diagnosis and prognosis. Their adaptability is particularly evident in bioinformatics, where they are widely used for tasks such as protein sequence alignment and gene prediction. However, the high computational complexity of HMMs is a significant drawback. The model's reliance on accurate calculations and precise inputs can limit its computational efficiency. In certain applications, such as unit selection in speech synthesis, HMMs can be overly restrictive due to their single-path output, thereby limiting their ability to generate variation which is often desirable in tasks that require expressive output [42].

HMM with and without SMOTE modeling in this study uses the Gaussian HMM library from hmmlearn, which tests the number of hidden states from 1 to 10 and iterations from 1 to 5. To ensure that the research results obtained were optimal, several parameter adjustment experiments were carried out so that the covariance type used was 'diag', the decoder algorithm used was Viterbi, and the random state was 1.

F. EVALUATION

1. CONFUSION MATRIX

Confusion Matrix is one of the classic decision measurement methods in supervised machine learning. This matrix visualizes the degree of confusion of algorithms in different classifications and is independent of a particular classification algorithm. The columns of the confusion matrix represent the predicted class results, and the rows represent the actual class results [43].

Confusion Matrix is defined as a matrix that provides a mixture of predicted and actual class instances. It allows the definition of various performance metrics (e.g., accuracy, precision, gain, Mathews correlation coefficient, etc.) or techniques such as Receiver Operating Characteristic (ROC) and Area Under Curve (AUC). In binary classification, all performance metrics and ROC analysis can be applied. There is a set of performance metrics that can be applied to the confusion matrix of a classification problem to assess an algorithm or compare the performance of different algorithms. The confusion matrix for binary classification is presented in TABLE 4. Each column of the matrix represents an instance of the predicted class, while each row represents an instance of the actual class. The confusion matrix has a dimension of 2 x 2, where one label is considered 'Positive' and the other label is considered 'Negative'. The matrix elements are characterized by the predicted label (positive, negative) and the result of comparing the predicted class label with the actual class label (true false): True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [44].

TABLE 4
Confusion Matrix

Classification		Prediction	
		Positive	Negative
Actual	Positive	True Positive	True Negative
	Negative	False Positive	False Negative

The quality of the classifier is measured based on the confusion matrix. Measurements based on the confusion matrix include, but are not limited to, accuracy, precision, and recall. Accuracy written in Eq. (13) [29] indicates how often the classifier makes correct predictions; it is the ratio of the number of accurate predictions to the total number of predictions [29].

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{13}$$

Where TP means that samples from the actual class have been classified into the same predicted class. FN indicates that samples in the actual class have been classified into other predicted classes. FP indicates that samples from other real classes have been classified into the selected predicted class. TN indicates that, for the selected real class, samples from other actual classes were classified into predicted classes other than the predicted class corresponding to the chosen real class. Precision written in Eq. (14) [29] determines how many samples, out of all those classified as positive, are samples of the positive class.

$$Precision = \frac{TP}{TP+FP} \tag{14}$$

Recall written in Eq. (15) [29] is used to determine how many samples belonging to the positive class were classified as positive by the classifier.

$$Recall = \frac{TP}{TP+FN} \tag{15}$$

The accuracy of a classification model is an important metric for measuring how often the model correctly predicts the class or label of a given data point. Precision measures the proportion of correctly predicted positive examples out of all examples predicted as positive. And recall calculates the proportion of correctly predicted positive examples out of all true positive examples. Furthermore, precision and recall are critical to understanding the model's ability to minimize false positives and false negatives [45], [46].

2. AUC

Area Under Curve (AUC) is a standard method for calculating probabilities in mathematical statistics through linear summation for discrete variables or integration for continuous variables [47]. Area Under Curve (AUC), known as the AUC statistic or c statistic, is a quantitative measure of Receiver Operating Characteristic (ROC). ROC is a two-dimensional performance measure of a credit scoring model. The AUC measures the predictive accuracy of the credit scoring model. The larger the AUC, the more accurate the model. AUC is the sum of the areas of the triangle, rectangle, and trapezoid [48]. Although often used in research dealing with biological markers such as daily cortisol and ROC curves to identify cut-off points, AUC has recently been used in other research fields (e.g., dentistry and sleep research) [49].

The ROC curve (Receiver Operating Characteristic curve) plots the True Positive Rate (TPR) written in Eq. (16) [45] versus the False Positive Rate (FPR) written in Eq. (17) [45] at different classification thresholds.

$$TPR = \frac{\sum TP}{\sum TP + \sum FN} \quad (16)$$

$$FPR = \frac{\sum FP}{\sum TN + \sum FP} \quad (17)$$

AUC evaluates the model's ability to distinguish between classes by plotting the true positive rate against the false positive rate. The higher the AUC, the higher the TPR, which indicates that the model is performing well. Accuracy metrics can be misleading if the classes are imbalanced, meaning one class has more examples. AUC is a robust metric less affected by class imbalance and provides a comprehensive view of the model's performance across all thresholds [45].

TABLE 5
AUC Interpretation

Area Under The Curve (AUC)	Interpretation
0,9 <= AUC	Excellent
0,8 <= AUC <= 0,9	Good
0,7 <= AUC <= 0,8	Fair
0,6 <= AUC <= 0,7	Poor
0,5 <= AUC <= 0,6	Fail

AUC stands for "Area Under the ROC Curve". An ideal ROC curve thus has AUC = 1.0 [50].

For the diagnosis test to be more accurate, the AUC should be greater than 0.5. Generally, an AUC ≥ 0.8 is considered acceptable [51].

III. RESULTS

This section shows the performance of each model to detect recurrence of thyroid cancer, using Extreme Learning Machine, Extreme Learning Machine with SMOTE, Hidden Markov Model, and Hidden Markov Model with SMOTE.

The performance of each model is evaluated with accuracy, precision, recall, and AUC values.

A. EXTREME LEARNING MACHINE

In this study, the first classification is using the Extreme Machine Learning algorithm. The activation function used in this study is a Binary Sigmoid activation function, and The random state used is 42. Meanwhile the neurons tested in the hidden layer numbered 5 to 50 with a multiple of 5. The evaluation results of the ELM model performance can be observed in TABLE 6. Each data split ratio obtained an evaluation value for each hidden neuron tested.

The best evaluation results of classification with ELM based on each data division ratio are as follows:

- At a ratio of 90:10 get the best results on 45 neurons with an accuracy value of 1.00, precision 1.00, recall 1.00, and AUC 1.00.
- At a ratio of 80:20, the best results were obtained on hidden neurons 45 with an accuracy value of 0.974, precision of 0.9651, recall of 0.9651, and AUC of 0.9973.
- At a ratio of 70:30, the best results were obtained for hidden neuron 50, with an accuracy value of 0.9391, precision 0.9278, recall 0.9194, and AUC 0.9684.
- At a ratio of 60:40, the best results were obtained for hidden neuron 30, with an accuracy value of 0.9416, precision of 0.9238, recall of 0.9301, and AUC of 0.9973.
- At a ratio of 50:50, the best results were obtained on hidden neuron 45, with an accuracy value of 0.9531, precision of 0.9368, recall of 0.9424, and AUC of 0.9744.

TABLE 6
ELM Performance

ELM	Hidden Neuron										
	5	10	15	20	25	30	35	40	45	50	
90:10	Accuracy	0.6923	0.8462	0.9487	0.9487	0.9744	0.9744	0.9744	0.9744	1.0000	1.0000
	Precision	0.3553	0.8101	0.9367	0.9367	0.9828	0.9828	0.9828	0.9828	1.0000	1.0000
	Recall	0.4821	0.8101	0.9367	0.9367	0.9545	0.9545	0.9545	0.9545	1.0000	1.0000
	AUC	0.8929	0.9578	0.9870	0.9935	0.9935	0.9968	1.0000	1.0000	1.0000	1.0000
80:20	Accuracy	0.7922	0.8701	0.9091	0.9221	0.9351	0.9610	0.9610	0.9610	0.9740	0.9740
	Precision	0.7958	0.8435	0.8828	0.8952	0.9075	0.9553	0.9553	0.9553	0.9651	0.9651
	Recall	0.5966	0.7899	0.8689	0.8952	0.9215	0.9387	0.9387	0.9387	0.9651	0.9651
70:30	AUC	0.8630	0.9728	0.9256	0.9628	0.9837	0.9855	0.9964	0.9964	0.9973	0.9964
	Accuracy	0.7652	0.8522	0.8783	0.9043	0.9217	0.9304	0.9217	0.9217	0.9391	0.9391
	Precision	0.8082	0.8333	0.8590	0.8836	0.8999	0.9134	0.9134	0.9134	0.9278	0.9278
60:40	Recall	0.5877	0.7824	0.8293	0.8761	0.9074	0.9134	0.8882	0.8882	0.9194	0.9194
	AUC	0.8151	0.9608	0.9349	0.9721	0.9703	0.9849	0.9883	0.9774	0.9646	0.9684
	Accuracy	0.7597	0.8571	0.8896	0.9156	0.9286	0.9416	0.9221	0.9286	0.9351	0.9416
50:50	Precision	0.8061	0.8407	0.8719	0.9011	0.9077	0.9238	0.9068	0.9125	0.9182	0.9238
	Recall	0.5670	0.7827	0.8423	0.8824	0.9137	0.9301	0.8943	0.9063	0.9182	0.9301
	AUC	0.7725	0.9505	0.9428	0.9588	0.9711	0.9828	0.9619	0.9722	0.9779	0.9630
50:50	Accuracy	0.7708	0.8802	0.9115	0.9115	0.9427	0.9427	0.9427	0.9375	0.9531	0.9271
	Precision	0.8207	0.8524	0.8910	0.8805	0.9234	0.9234	0.9333	0.9189	0.9368	0.9097
	Recall	0.5665	0.8283	0.8754	0.8948	0.9289	0.9289	0.9159	0.9189	0.9424	0.8989
50:50	AUC	0.7556	0.9546	0.9393	0.9627	0.9675	0.9811	0.9666	0.9689	0.9744	0.9680

B. EXTREME LEARNING MACHINE WITH SMOTE

The second classification in this study is using the Extreme Learning Machine algorithm combined with SMOTE. The dataset used has imbalanced data, namely the class of recurrence after treatment with a percentage of 28% while the class of no recurrence after treatment with a percentage of 72%, so using the SMOTE method to deal with the problem of data imbalance. The SMOTE parameters used are k nearest neighbors of 5 and random state of 42, while the ELM modeling uses a Binary Sigmoid activation function and random state 43. The evaluation results of the ELM with SMOTE model performance can be observed in TABLE 7.

The best evaluation results of classification with ELM with SMOTE based on each data division ratio are as follows:

- At a ratio of 90:10 get the best results on 35 neurons with an accuracy value of 1.00, precision 1.00, recall 1.00, and AUC 1.00.
- At a ratio of 80:20, the best results were obtained on hidden neurons 35 with an accuracy value of 1.00, precision of 1.00, recall of 1.00, and AUC of 1.00.
- At a ratio of 70:30, the best results were obtained for hidden neuron 35, with an accuracy value of 0.9565, precision 0.9324, recall 0.9699, and AUC 0.9974.
- At a ratio of 60:40, the best results were obtained for hidden neuron 50, with an accuracy value of 0.9286, precision 0.8962, recall 0.9509, and AUC 0.9909.
- At a ratio 50:50, the best results are obtained at hidden neuron 45 with an accuracy value of 0.9167, precision 0.8789, recall 0.9372, and AUC 0.9856.

TABLE 7

ELM + SMOTE Performance

ELM + SMOTE		Hidden Neuron									
		5	10	15	20	25	30	35	40	45	50
90:10	Accuracy	0.5897	0.8205	0.8718	0.8718	0.8974	0.8718	1.0000	1.0000	0.9744	1.0000
	Precision	0.6332	0.7908	0.8371	0.8371	0.8734	0.8483	1.0000	1.0000	0.9828	1.0000
	Recall	0.6591	0.8474	0.8831	0.8831	0.8734	0.8279	1.0000	1.0000	0.9545	1.0000
	AUC	0.8247	0.9545	0.9383	0.9416	0.9838	0.9805	1.0000	1.0000	1.0000	1.0000
80:20	Accuracy	0.4545	0.7792	0.8961	0.8831	0.9610	0.9610	1.0000	1.0000	1.0000	1.0000
	Precision	0.5582	0.7531	0.8504	0.8364	0.9318	0.9412	1.0000	1.0000	1.0000	1.0000
	Recall	0.5672	0.8358	0.9133	0.8693	0.9741	0.9564	1.0000	1.0000	1.0000	1.0000
	AUC	0.7069	0.9564	0.9655	0.9229	0.9946	0.9973	1.0000	1.0000	1.0000	1.0000
70:30	Accuracy	0.5391	0.7739	0.8609	0.8783	0.9217	0.9217	0.9565	0.9478	0.9565	0.9478
	Precision	0.6175	0.7614	0.8268	0.8425	0.8909	0.8909	0.9324	0.9211	0.9324	0.9242
	Recall	0.6327	0.8242	0.8844	0.8869	0.9362	0.9362	0.9699	0.9639	0.9699	0.9543
	AUC	0.7195	0.9239	0.9571	0.9420	0.9853	0.9921	0.9974	0.9951	0.9944	0.9872
60:40	Accuracy	0.5325	0.7273	0.8377	0.8766	0.8961	0.8896	0.9286	0.9221	0.9221	0.9286
	Precision	0.6188	0.7303	0.7988	0.8394	0.8610	0.8535	0.8962	0.8889	0.8889	0.8962
	Recall	0.6339	0.7902	0.8512	0.8929	0.9211	0.9092	0.9509	0.9464	0.9464	0.9509
	AUC	0.7277	0.9014	0.9237	0.9392	0.9841	0.9753	0.9824	0.9841	0.9855	0.9909
50:50	Accuracy	0.5208	0.6927	0.7865	0.8594	0.8854	0.8750	0.8802	0.9010	0.8958	0.9167
	Precision	0.5925	0.7004	0.7488	0.8186	0.8472	0.8347	0.8409	0.8617	0.8557	0.8789
	Recall	0.6113	0.7599	0.8103	0.8855	0.9225	0.9025	0.9125	0.9266	0.9166	0.9372
	AUC	0.7431	0.8830	0.9104	0.9407	0.9818	0.9690	0.9687	0.9766	0.9738	0.9856

C. HIDDEN MARKOV MODEL

The third classification in this study is using the Hidden Markov Model algorithm. The covariance type used is diag, the decoder algorithm used is the Viterbi algorithm, and random state of 1. Meanwhile the parameters tested in this study are the number of hidden states and the number of iterations, hidden states are 1 to 10, and iteration are 1 to 5. The evaluation results of the HMM model performance can be observed in TABLE 8.

The best evaluation results of classification with HMM based on each data division ratio are as follows:

- At a ratio of 90:10 get the best results on hidden state 1 and 1 iteration with an accuracy value of 0.7179, precision 0.3590, recall 0.5, and AUC 0.5.

- At a ratio of 80:20, get the best results on hidden state 2 and 3 iterations with an accuracy value of 0.8571, precision 0.8474, recall 0.7459, and AUC 0.96.
- At a ratio of 70:30, get the best results on hidden state 1 and 1 iteration with an accuracy value of 0.7217, precision 0.3608, recall 0.5, and AUC 0.5.
- At a ratio of 60:40, get the best results in hidden state 2 and 3 iterations with an accuracy value of 0.8636, precision 0.8471, recall 0.7946, and AUC 0.9343.
- At a ratio of 50:50, get the best results on hidden state 1 and 1 iteration with an accuracy value of 0.7395, precision 0.3697, recall 0.5, and AUC 0.5.

TABLE 8
HMM Performance with 5 iterations

Hidden State	Accuracy					Precision					Recall					AUC				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
90:10																				
1	0.72	0.72	0.72	0.72	0.72	0.36	0.36	0.36	0.36	0.36	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2	0.13	0.15	0.15	0.18	0.18	0.13	0.18	0.18	0.22	0.22	0.20	0.22	0.24	0.24	0.08	0.17	0.18	0.18	0.18	0.21
3	0.51	0.46	0.44	0.44	0.44	0.57	0.57	0.56	0.56	0.56	0.33	0.31	0.29	0.29	0.29	0.95	0.94	0.94	0.95	0.96
...																				...
8	0.03	0.05	0.05	0.05	0.05	0.13	0.25	0.19	0.19	0.19	0.01	0.02	0.02	0.02	0.02	0.66	0.55	0.55	0.55	0.55
9	0.15	0.28	0.28	0.21	0.21	0.11	0.11	0.13	0.11	0.11	0.02	0.04	0.05	0.03	0.03	0.14	0.13	0.13	0.13	0.21
10	0.08	0.08	0.08	0.10	0.10	0.10	0.10	0.10	0.11	0.11	0.01	0.01	0.01	0.02	0.02	0.85	0.59	0.59	0.59	0.59
80:20																				
1	0.75	0.75	0.75	0.75	0.75	0.38	0.38	0.38	0.38	0.38	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2	0.86	0.86	0.86	0.86	0.86	0.85	0.85	0.85	0.85	0.85	0.75	0.75	0.75	0.75	0.75	0.96	0.96	0.96	0.96	0.96
3	0.04	0.06	0.06	0.06	0.06	0.04	0.11	0.10	0.10	0.10	0.05	0.06	0.06	0.06	0.06	0.29	0.27	0.26	0.22	0.18
...																				...
8	0.01	0.04	0.06	0.08	0.08	0.04	0.10	0.11	0.13	0.13	0.00	0.02	0.03	0.04	0.04	0.31	0.37	0.46	0.46	0.46
9	0.06	0.04	0.01	0.01	0.01	0.17	0.17	0.06	0.06	0.06	0.01	0.01	0.00	0.00	0.00	0.34	0.41	0.60	0.68	0.66
10	0.04	0.05	0.05	0.04	0.04	0.15	0.17	0.15	0.10	0.10	0.01	0.01	0.01	0.01	0.01	0.70	0.68	0.62	0.65	0.62
70:30																				
1	0.72	0.72	0.72	0.72	0.72	0.36	0.36	0.36	0.36	0.36	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2	0.15	0.14	0.14	0.14	0.14	0.16	0.15	0.15	0.15	0.15	0.23	0.21	0.21	0.21	0.21	0.11	0.20	0.20	0.20	0.20
3	0.08	0.06	0.07	0.08	0.08	0.11	0.10	0.10	0.11	0.11	0.07	0.05	0.06	0.06	0.06	0.29	0.26	0.24	0.24	0.24
...																				...
8	0.13	0.12	0.10	0.11	0.12	0.18	0.16	0.15	0.17	0.18	0.03	0.04	0.04	0.04	0.04	0.44	0.45	0.50	0.50	0.50
9	0.10	0.08	0.05	0.07	0.08	0.22	0.22	0.22	0.19	0.22	0.03	0.02	0.02	0.02	0.02	0.87	0.72	0.59	0.59	0.59
10	0.10	0.17	0.17	0.10	0.11	0.09	0.09	0.09	0.10	0.10	0.02	0.02	0.03	0.02	0.02	0.14	0.29	0.36	0.36	0.36
60:40																				
1	0.73	0.73	0.73	0.73	0.73	0.36	0.36	0.36	0.36	0.36	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2	0.86	0.86	0.86	0.86	0.86	0.84	0.83	0.85	0.84	0.84	0.78	0.79	0.79	0.78	0.78	0.93	0.94	0.93	0.93	0.93
3	0.08	0.09	0.10	0.10	0.11	0.10	0.15	0.21	0.21	0.21	0.09	0.09	0.10	0.10	0.11	0.34	0.40	0.40	0.42	0.43
...																				...
8	0.12	0.10	0.10	0.09	0.14	0.21	0.21	0.19	0.17	0.20	0.03	0.03	0.03	0.02	0.03	0.56	0.56	0.51	0.52	0.51
9	0.04	0.05	0.05	0.05	0.03	0.12	0.13	0.16	0.17	0.17	0.01	0.01	0.01	0.01	0.01	0.22	0.19	0.24	0.17	0.16
10	0.25	0.27	0.27	0.36	0.37	0.12	0.12	0.13	0.15	0.15	0.04	0.04	0.04	0.05	0.05	0.18	0.19	0.29	0.29	0.29
50:50																				
1	0.74	0.74	0.74	0.74	0.74	0.37	0.37	0.37	0.37	0.37	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2	0.14	0.14	0.14	0.14	0.14	0.16	0.17	0.16	0.16	0.16	0.22	0.21	0.21	0.21	0.21	0.14	0.16	0.19	0.18	0.18
3	0.46	0.42	0.29	0.27	0.28	0.33	0.30	0.27	0.27	0.28	0.24	0.21	0.14	0.13	0.14	0.22	0.20	0.18	0.17	0.17
...																				...
8	0.06	0.06	0.07	0.08	0.08	0.10	0.07	0.09	0.12	0.12	0.01	0.01	0.01	0.02	0.02	0.57	0.50	0.51	0.55	0.57
9	0.05	0.04	0.07	0.06	0.07	0.17	0.11	0.20	0.22	0.19	0.02	0.02	0.02	0.02	0.03	0.67	0.74	0.75	0.65	0.65
10	0.07	0.09	0.11	0.13	0.13	0.09	0.09	0.10	0.10	0.10	0.01	0.02	0.02	0.02	0.02	0.29	0.37	0.40	0.41	0.42

D. HIDDEN MARKOV MODEL WITH SMOTE

The fourth classification in this study is using the Hidden Markov Model algorithm by combining SMOTE. The SMOTE parameters used are k-nearest neighbors of 4 and random states of 42, while the HMM modeling uses .diag for covariance type, Viterbi algorithm for decoder algorithm, and random state of 1. The evaluation results of the HMM with SMOTE model performance can be observed in **TABLE 9**.

The best evaluation results of classification with HMM with SMOTE based on each data division ratio are as follows:

- At a ratio of 90:10 get the best results on hidden state 2 and 4 iterations with an accuracy value of 0.8461, precision 0.8504, recall 0.7548, and AUC 0.9594.

- At a ratio of 80:20, get the best results on hidden state 2 and 2 iterations with an accuracy value of 0.8312, precision 0.8505, recall 0.6756, and AUC 0.8884.
- At a ratio of 70:30, get the best results on hidden state 2 and 2 iterations with an accuracy value of 0.8696, precision 0.8832, recall 0.7848, and AUC 0.9174.
- At a ratio of 60:40, get the best results on hidden state 1 and 1 iteration with an accuracy value of 0.7273, precision of 0.3636, recall 0.5, and AUC 0.5.
- At a ratio of 50:50 get the best results on hidden state 1 and 1 iteration with an accuracy value of 0.7395, precision 0.3697, recall 0.5, and AUC 0.5.

TABLE 9
HMM + SMOTE Performance with 5 iterations

Hidden State	Accuracy					Precision					Recall					AUC				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
90:10																				
1	0.72	0.72	0.72	0.72	0.72	0.36	0.36	0.36	0.36	0.36	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
2	0.82	0.82	0.82	0.85	0.85	0.83	0.83	0.83	0.85	0.85	0.71	0.71	0.71	0.75	0.75	0.95	0.96	0.96	0.96	0.96
3	0.49	0.46	0.44	0.44	0.44	0.57	0.57	0.54	0.54	0.54	0.32	0.31	0.28	0.28	0.28	0.96	0.96	0.96	0.90	0.90
...																				...
8	0.05	0.05	0.03	0.03	0.05	0.07	0.06	0.04	0.04	0.06	0.01	0.01	0.00	0.00	0.01	0.45	0.48	0.39	0.37	0.36
9	0.08	0.08	0.08	0.08	0.08	0.22	0.17	0.15	0.15	0.15	0.02	0.03	0.02	0.02	0.02	0.80	0.68	0.68	0.68	0.67
10	0.03	0.03	0.03	0.05	0.05	0.04	0.05	0.06	0.07	0.07	0.00	0.00	0.00	0.01	0.01	0.11	0.14	0.21	0.21	0.21

TABLE 9
 (Continued)

Hidden State	Accuracy					Precision					Recall					AUC				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
80:20																				
1	0,75	0,75	0,75	0,75	0,75	0,38	0,38	0,38	0,38	0,38	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50
2	0,83	0,83	0,83	0,83	0,83	0,85	0,85	0,85	0,85	0,85	0,68	0,68	0,68	0,68	0,68	0,89	0,89	0,89	0,89	0,89
3	0,8	0,8	0,8	0,8	0,8	0,5	0,5	0,5	0,5	0,5	0,4	0,4	0,4	0,4	0,4	0,5	0,6	0,6	0,6	0,6
...																				
8	0,4	0,4	0,4	0,4	0,4	0,1	0,2	0,2	0,2	0,2	0,1	0,1	0,1	0,1	0,1	0,7	0,6	0,6	0,6	0,6
9	0,2	0,4	0,3	0,4	0,4	0,3	0,2	0,3	0,3	0,3	0	0,1	0,1	0,1	0,1	0,8	0,7	0,6	0,6	0,6
10	0,5	0,4	0,2	0,1	0,1	0,2	0,2	0,2	0,2	0,2	0,1	0,1	0	0	0	0,8	0,6	0,6	0,6	0,6
70:30																				
1	0,72	0,72	0,72	0,72	0,72	0,36	0,36	0,36	0,36	0,36	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50
2	0,87	0,87	0,86	0,85	0,85	0,88	0,88	0,86	0,84	0,84	0,78	0,78	0,78	0,77	0,77	0,89	0,92	0,91	0,90	0,90
3	0,63	0,57	0,55	0,55	0,55	0,58	0,55	0,54	0,53	0,53	0,41	0,35	0,32	0,32	0,32	0,94	0,95	0,87	0,87	0,87
...																				
8	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,2	0,2	0,2	0	0	0	0	0	0,7	0,7	0,7	0,8	0,8
9	0	0	0	0	0	0,1	0,1	0,1	0,1	0,1	0	0	0	0	0	0,6	0,6	0,6	0,6	0,6
10	0,1	0,1	0	0	0	0,1	0,1	0,1	0,1	0,1	0	0	0	0	0	0,2	0,3	0,3	0,3	0,3
60:40																				
1	0,73	0,73	0,73	0,73	0,73	0,36	0,36	0,36	0,36	0,36	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50
2	0,1	0,2	0,2	0,2	0,2	0,1	0,1	0,1	0,1	0,1	0,3	0,3	0,3	0,3	0,3	0,2	0,2	0,2	0,2	0,2
3	0,6	0,6	0,6	0,6	0,6	0,6	0,5	0,5	0,6	0,5	0,4	0,4	0,4	0,4	0,3	0,9	0,9	0,9	0,9	0,9
...																				
8	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0	0	0	0	0	0,4	0,5	0,5	0,5	0,5
9	0	0	0	0	0	0,1	0,1	0,1	0,1	0,1	0	0	0	0	0	0,4	0,5	0,5	0,6	0,6
10	0	0	0	0	0	0,1	0	0,1	0,1	0	0	0	0	0	0	0,4	0,4	0,4	0,4	0,4
50:50																				
1	0,74	0,74	0,74	0,74	0,74	0,37	0,37	0,37	0,37	0,37	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50	0,50
2	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2
3	0,6	0,7	0,7	0,7	0,7	0,9	0,9	0,9	0,9	0,9	0,6	0,6	0,6	0,6	0,6	0,4	0,4	0,4	0,4	0,4
...																				
8	0,1	0,1	0,1	0,1	0	0,2	0,1	0,1	0,2	0,1	0	0	0	0	0	0,6	0,7	0,7	0,7	0,7
9	0,1	0,1	0	0,1	0	0,1	0,2	0,2	0,2	0,2	0	0	0	0	0	0,6	0,6	0,7	0,7	0,7
10	0,4	0,4	0,3	0,3	0,3	0,2	0,2	0,2	0,2	0,2	0,1	0,1	0	0,1	0,1	0,6	0,6	0,6	0,6	0,6

IV. DISCUSSION

Based on the research results previously explained, the best results are identified from the four models performed, which are ELM, ELM with SMOTE, HMM, and HMM with SMOTE. A comparison of the highest performance of all modeling can be seen in FIGURE 5.

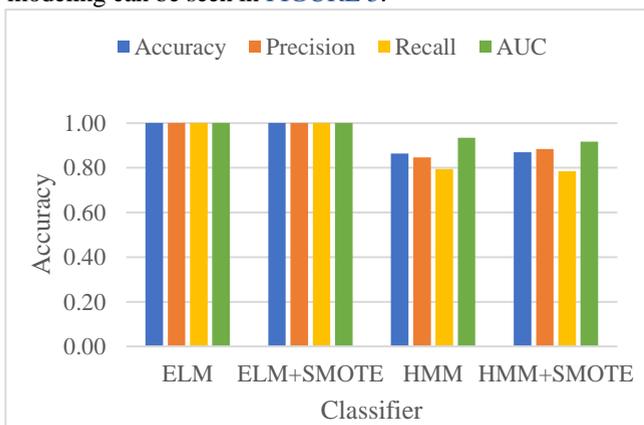


FIGURE 5. Comparison of Each Model

The best results in the ELM model are found in the 90:10 ratio with 45 hidden neurons, while the best results in ELM modeling with SMOTE are found in the 90:10 ratio with 35 hidden neurons. The best results in HMM modeling are in the 60:40 ratio with 2 hidden states and 3 iterations, while the best

results in HMM modeling with SMOTE are in the 70:30 ratio with 2 hidden states and 2 iterations.

Based on the research results, the application of SMOTE to ELM and HMM improves performance results in most experiments. It can be seen that the ELM test with SMOTE at a ratio of 90:10 with 35 hidden neurons resulted in an accuracy value of 1.00, precision of 1.00, recall of 1.00, and AUC of 1.00. In comparison, the ELM test without SMOTE at the same ratio and number of hidden neurons resulted in an accuracy value of 0.9744, precision of 0.9828, recall of 0.9545, and AUC of 1.00.

It can also be seen in testing HMM with SMOTE at a ratio of 90:10 with 2 hidden states and 4 iterations, resulting in an accuracy value of 0.8461, precision of 0.8504, recall of 0.7548, and AUC of 0.9594. The HMM test without SMOTE at the same ratio, number of hidden states, and iterations resulted in an accuracy value of 0.1794, precision 0.2189, recall 0.2353, and AUC 0.1753.

Based on the ROC-AUC curve in FIGURE 6 of the highest values of each modeling, all models performed very well: All ROC curves are very close to the upper left corner of the graph, which indicates high TPR and low FPR. This indicates that these four models can classify instances with high accuracy. ELM and ELM with SMOTE performed the best: Both models have an AUC of 1.00, indicating that they distinguish between positive and negative classes.

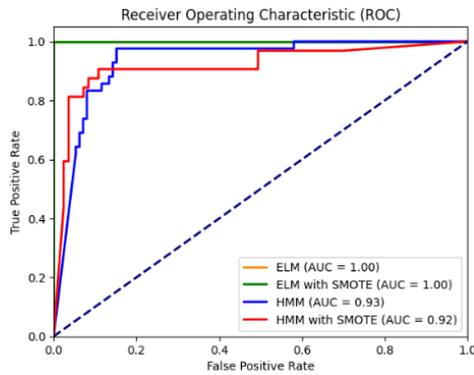


FIGURE 6. ROC-AUC Curve of highest values of each model

Research with the dataset 'Recurrence of Differentiated Thyroid Cancer' was conducted by [9] by comparing several machine learning methods such as Logistic Regression, Naïve Bayes, Decision Tree, and KNN. A comparison of model performance from the previous study with this study can be seen in TABLE 10.

TABLE 10
 Comparison with Previous Research on Recurrence of Differentiated Thyroid Cancer

Research	Classifier	Accuracy
[9]	LR	0.91
	NB	0.86
	DT	0.91
	KNN	0.90
Proposed Work	ELM	1.00
	ELM-SMOTE	1.00
	HMM	0.86
	HMM-SMOTE	0.87

A comparison of the performance results of previous research with this study with the same machine learning method can be seen in TABLE 11.

TABLE 11
 Comparison of Model with Previous Research

Research	Classifier	Dataset	Accuracy
[52]	ELM	Lung Cancer	0.89
	ELM-SMOTE		0.85
[53]	ELM	Hydraulic Pump	1.00
[40]	HMM	Parkinson's Disease	0.96
		ELM	1.00
Proposed Work	ELM-SMOTE	Recurrence of Differentiated Thyroid Cancer	1.00
	HMM	0.86	
	HMM-SMOTE	0.87	

Based on the results and explanations above, it can be concluded that the Extreme Learning Machine algorithm, especially ELM with SMOTE, is a better algorithm than the Hidden Markov Model in predicting recurrence of differentiated thyroid cancer. In addition, the application of SMOTE has been proven to improve the performance of ELM and HMM modeling in predicting the recurrence of differentiated thyroid cancer.

Meanwhile, the weakness of the research results of this study lies in HMM modeling, where the more hidden states are tested, the lower the accuracy value. This is because HMM is an algorithm that has high complexity so that when the

number of hidden states is less, the model tends to be simpler, while models with many hidden states make the modeling more complex. To address this, future research is strongly recommended to focus on dimension reduction techniques and extracting relevant features to reduce the model's complexity.

The implications of this research are extensive in medical prediction and machine learning. Using ELM with SMOTE significantly improved the accuracy in predicting the recurrence of differentiated thyroid cancer compared to other models. This means that this model can be more reliable in identifying patients at high risk of recurrence. SMOTE increases the number of minority samples (i.e., patients who experience recurrence), which helps the ELM model be more sensitive in detecting positive cases. This study demonstrates the effectiveness of SMOTE in addressing the problem of class imbalance in medical data, which is often a challenge in developing prediction models. Using ELM can speed up the diagnosis and medical decision-making process in clinical practice. The ELM model with SMOTE can be a valuable tool for researchers and medical practitioners to improve treatment outcomes for differentiated thyroid cancer patients. HMMs, although more complex, may be more suitable for applications that require detailed modeling of temporal dynamics, such as monitoring disease progression over time. Overall, this study's results provide practical guidance for researchers and practitioners in selecting and implementing machine learning models for differentiated thyroid cancer recurrence prediction and underscore the importance of customizing and validating models according to the data characteristics used.

V. CONCLUSION

This research uses machine learning methods to detect the recurrence of differentiated thyroid cancer. This research begins with collecting datasets from Kaggle, preprocessing data with encoding methods, dividing data into several ratio proportions, balancing data with SMOTE, modeling ELM and HMM data, and analyzing performance results with accuracy, precision, recall, and AUC. This research conducted four machine learning models: ELM, ELM with SMOTE, HMM, and HMM with SMOTE.

ELM modeling gets the best results at a ratio of 90:10 with 45 hidden neurons that get an accuracy value of 1.00, precision 1.00, recall 1.00, and AUC 1.00. ELM modeling with SMOTE gets the best results at a ratio of 90:10 with 35 hidden neurons that get an accuracy value of 1.00, precision 1.00, recall 1.00, and AUC 1.00. HMM modeling gets the best value at a ratio of 60:40 with 2 hidden states and 3 iterations, which get an accuracy value of 0.8636, precision 0.8471, recall 0.7946, and AUC 0.9343. Last, HMM modeling with SMOTE gets the best results at a ratio of 70:30 with 2 hidden states and 2 iterations, which get an accuracy value of 0.8696, precision 0.8832, recall 0.7848, and AUC 0.9174.

Based on the highest result of each modeling, ELM with SMOTE was shown to have better performance than others model, then ELM without SMOTE, and then HMM with SMOTE, last HMM without SMOTE. ELM work better than

HMM in the prediction recurrence of differentiated thyroid cancer. ELM's ability to generalise is better than HMM, and the application of SMOTE to ELM and HMM is proven to improve performance results in most experiments.

This research still has weaknesses, which is the low accuracy value on more of the hidden states in HMM testing. This is because HMM is an algorithm with high complexity. Thus, future research is recommended to use dimension reduction or feature extraction techniques such as Principal Component Analysis (PSA) or similar methods to reduce model complexity.

REFERENCES

- [1] F. Bray *et al.*, "Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA Cancer J Clin*, vol. 74, no. 3, pp. 229–263, May 2024, doi: 10.3322/caac.21834.
- [2] C. Cortas and H. Charalambous, "Tyrosine Kinase Inhibitors for Radioactive Iodine Refractory Differentiated Thyroid Cancer," *Life*, vol. 14, no. 1, p. 22, Dec. 2023, doi: 10.3390/life14010022.
- [3] A. M. Bulfamante *et al.*, "Advanced Differentiated Thyroid Cancer: A Complex Condition Needing a Tailored Approach," *Front Oncol*, vol. 12, Jul. 2022, doi: 10.3389/fonc.2022.954759.
- [4] M. Pacilio *et al.*, "Personalized Dosimetry in the Context of Radioiodine Therapy for Differentiated Thyroid Cancer," Jul. 01, 2022, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/diagnostics12071763.
- [5] H. R. Won *et al.*, "Is Maintaining Thyroid-Stimulating Hormone Effective in Patients Undergoing Thyroid Lobectomy for Low-Risk Differentiated Thyroid Cancer? A Systematic Review and Meta-Analysis," Mar. 01, 2022, *MDPI*. doi: 10.3390/cancers14061470.
- [6] S. De Leo *et al.*, "Post-surgical ablative or adjuvant radioiodine therapy has no impact on outcome in 1–4 cm differentiated thyroid cancers without extrathyroidal extension," *J Clin Med*, vol. 10, no. 19, Oct. 2021, doi: 10.3390/jcm10194452.
- [7] A. Kuang, V. L. Kouznetsova, S. Kesari, and I. F. Tsigelny, "Diagnostics of Thyroid Cancer Using Machine Learning and Metabolomics," *Metabolites*, vol. 14, no. 1, Jan. 2024, doi: 10.3390/metabo14010011.
- [8] J. A. Chandio, G. A. Mallah, and N. A. Shaikh, "Decision Support System for Classification Medullary Thyroid Cancer," *IEEE Access*, vol. 8, pp. 145216–145226, 2020, doi: 10.1109/ACCESS.2020.3014863.
- [9] K. E. Setiawan, "PREDICTING RECURRENCE IN DIFFERENTIATED THYROID CANCER: A COMPARATIVE ANALYSIS OF VARIOUS MACHINE LEARNING MODELS INCLUDING ENSEMBLE METHODS WITH CHI-SQUARED FEATURE SELECTION," *Communications in Mathematical Biology and Neuroscience*, vol. 2024, 2024, doi: 10.28919/cmbn/8506.
- [10] E. J. Sadgrove, G. Falzon, D. Miron, and D. W. Lamb, "The segmented colour feature extreme learning machine: Applications in agricultural robotics," *Agronomy*, vol. 11, no. 11, Nov. 2021, doi: 10.3390/agronomy11112290.
- [11] N. Bacanin, C. Stoean, M. Zivkovic, D. Jovanovic, M. Antonijevic, and D. Mladenovic, "Multi-Swarm Algorithm for Extreme Learning Machine Optimization," *Sensors*, vol. 22, no. 11, Jun. 2022, doi: 10.3390/s22114204.
- [12] V. Lahoura *et al.*, "Cloud computing-based framework for breast cancer diagnosis using extreme learning machine," *Diagnostics*, vol. 11, no. 2, Feb. 2021, doi: 10.3390/diagnostics11020241.
- [13] U. Ayman *et al.*, "Epileptic Patient Activity Recognition System Using Extreme Learning Machine Method," *Biomedicines*, vol. 11, no. 3, Mar. 2023, doi: 10.3390/biomedicines11030816.
- [14] B. Mor, S. Garhwal, and A. Kumar, "A Systematic Review of Hidden Markov Models and Their Applications," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1429–1448, May 2021, doi: 10.1007/s11831-020-09422-4.
- [15] Y. Wang, C. Wang, H. Wang, and Z. Chen, "Prediction of Consumers' Adoption Behavior of Products with Water Efficiency Labeling Based on Hidden Markov Model," *Water (Switzerland)*, vol. 16, no. 1, Jan. 2024, doi: 10.3390/w16010044.
- [16] A. M. A. Alqadasi *et al.*, "Rule-Based Embedded HMMs Phoneme Classification to Improve Qur'anic Recitation Recognition," *Electronics (Switzerland)*, vol. 12, no. 1, Jan. 2023, doi: 10.3390/electronics12010176.
- [17] B. Priambodo, A. Ahmad, and R. A. Kadir, "Predicting Traffic Flow Propagation Based on Congestion at Neighbouring Roads Using Hidden Markov Model," *IEEE Access*, vol. 9, pp. 85933–85946, 2021, doi: 10.1109/ACCESS.2021.3075911.
- [18] S. Gholampour, "Impact of Nature of Medical Data on Machine and Deep Learning for Imbalanced Datasets: Clinical Validity of SMOTE Is Questionable," *Mach Learn Knowl Extr*, vol. 6, no. 2, pp. 827–841, Apr. 2024, doi: 10.3390/make6020039.
- [19] H. Karamti *et al.*, "Improving Prediction of Cervical Cancer Using KNN Imputed SMOTE Features and Multi-Model Ensemble Learning Approach," *Cancers (Basel)*, vol. 15, no. 17, Sep. 2023, doi: 10.3390/cancers15174412.
- [20] G. Ryan, P. Katarina, and D. Suhartono, "MBTI Personality Prediction Using Machine Learning and SMOTE for Balancing Data Based on Statement Sentences," *Information (Switzerland)*, vol. 14, no. 4, Apr. 2023, doi: 10.3390/info14040217.
- [21] X. Larriva-Novo, V. A. Villagr a, M. Vega-Barbas, D. Rivera, and M. Sanz Rodrigo, "An IoT-focused intrusion detection system approach based on preprocessing characterization for cybersecurity datasets," *Sensors (Switzerland)*, vol. 21, no. 2, pp. 1–15, Jan. 2021, doi: 10.3390/s21020656.
- [22] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks," *J Big Data*, vol. 7, no. 1, Dec. 2020, doi: 10.1186/s40537-020-00305-w.
- [23] M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding," *IEEE Access*, vol. 9, pp. 114381–114391, 2021, doi: 10.1109/ACCESS.2021.3104357.
- [24] J. Fonseca, G. Douzas, and F. Bacao, "Improving imbalanced land cover classification with k-means smote: Detecting and oversampling distinctive minority spectral signatures," *Information (Switzerland)*, vol. 12, no. 7, Jul. 2021, doi: 10.3390/info12070266.
- [25] F. Duan, S. Zhang, Y. Yan, and Z. Cai, "An Oversampling Method of Unbalanced Data for Mechanical Fault Diagnosis Based on MeanRadius-SMOTE," *Sensors*, vol. 22, no. 14, Jul. 2022, doi: 10.3390/s22145166.
- [26] M. K. Suryadi, R. Herteno, S. W. Saputro, M. R. Faisal, and R. A. Nugroho, "A Comparative Study of Various Hyperparameter Tuning on Random Forest Classification with SMOTE and Feature Selection Using Genetic Algorithm in Software Defect Prediction," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 2, pp. 137–147, Apr. 2024, doi: 10.35882/jeeemi.v6i2.375.
- [27] Putri Nabella, Rudy Herteno, Setyo Wahyu Saputro, Mohammad Reza Faisal, and Friska Abadi, "Impact of a Synthetic Data Vault for Imbalanced Class in Cross-Project Defect Prediction," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 2, pp. 219–230, Apr. 2024, doi: 10.35882/jeeemi.v6i2.409.
- [28] A. M. Akbar, R. Herteno, S. W. Saputro, M. R. Faisal, and R. A. Nugroho, "Enhancing Software Defect Prediction through Hybrid Optimization for Feature Selection and Gradient Boosting Classification," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 2, pp. 169–181, Apr. 2024, doi: 10.35882/jeeemi.v6i2.388.
- [29] J. Kozak, B. Probiez, K. Kania, and P. Juszczuk, "Preference-Driven Classification Measure," *Entropy*, vol. 24, no. 4, Apr. 2022, doi: 10.3390/e24040531.
- [30] F. Mercaldo, L. Brunese, F. Martinelli, A. Santone, and M. Cesarelli, "Experimenting with Extreme Learning Machine for Biomedical Image Classification," *Applied Sciences (Switzerland)*, vol. 13, no. 14, Jul. 2023, doi: 10.3390/app13148558.
- [31] N. Surantha and I. D. Gozali, "Evaluation of the Improved Extreme Learning Machine for Machine Failure Multiclass Classification," *Electronics (Switzerland)*, vol. 12, no. 16, Aug. 2023, doi: 10.3390/electronics12163501.

- [32] Q. Fang, W. Zhang, and X. Wang, "Visual navigation using inverse reinforcement learning and an extreme learning machine," *Electronics (Switzerland)*, vol. 10, no. 16, Aug. 2021, doi: 10.3390/electronics10161997.
- [33] Y. H. Choi, A. Sadollah, and J. H. Kim, "Improvement of Cyber-Attack Detection Accuracy from Urban Water Systems Using Extreme Learning Machine," *Applied Sciences*, vol. 10, no. 22, p. 8179, Nov. 2020, doi: 10.3390/app10228179.
- [34] B. Deng, X. Zhang, W. Gong, and D. Shang, "An overview of extreme learning machine," in *Proceedings - 2019 4th International Conference on Control, Robotics and Cybernetics, CRC 2019*, Institute of Electrical and Electronics Engineers Inc., Sep. 2019, pp. 189–195. doi: 10.1109/CRC.2019.00046.
- [35] A. Borucka, E. Kozłowski, R. Parczewski, K. Antosz, L. Gil, and D. Pieniak, "Supply Sequence Modelling Using Hidden Markov Models," *Applied Sciences (Switzerland)*, vol. 13, no. 1, Jan. 2023, doi: 10.3390/app13010231.
- [36] A. Jandera and T. Skovranek, "Customer Behaviour Hidden Markov Model," *Mathematics*, vol. 10, no. 8, Apr. 2022, doi: 10.3390/math10081230.
- [37] T. S. Ajani, A. L. Imoize, and A. A. Atayero, "An overview of machine learning within embedded and mobile devices-optimizations and applications," Jul. 01, 2021, *MDPI AG*. doi: 10.3390/s21134412.
- [38] P. Chen, D. Yi, and C. Zhao, "Trading strategy for market situation estimation based on hidden Markov model," *Mathematics*, vol. 8, no. 7, Jul. 2020, doi: 10.3390/math8071126.
- [39] M. Orchard, C. Muñoz-Poblete, J. I. Huircan, P. Galeas, and H. Rozas, "Harvest stage recognition and potential fruit damage indicator for berries based on hidden Markov models and the viterbi algorithm," *Sensors (Switzerland)*, vol. 19, no. 20, Oct. 2019, doi: 10.3390/s19204421.
- [40] K. Safi, W. H. F. Aly, H. Kanj, T. Khalifa, M. Ghedira, and E. Hutin, "Hidden Markov Model for Parkinson's Disease Patients Using Balance Control Data," *Bioengineering*, vol. 11, no. 1, Jan. 2024, doi: 10.3390/bioengineering11010088.
- [41] S. Ma, P. Wang, and H. Lee, "An Enhanced Hidden Markov Model for Map-Matching in Pedestrian Navigation," *Electronics (Switzerland)*, vol. 13, no. 9, pp. 1–27, 2024, doi: <https://doi.org/10.3390/electronics13091685>.
- [42] S. C. Dimri, R. Indu, H. S. Negi, N. Panwar, and M. Sarda, "Hidden Markov Model - Applications, Strengths, and Weaknesses," in *2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*, Dehradun, India: IEEE, Mar. 2024, pp. 300–305.
- [43] J. Xu, Y. Zhang, and D. Miao, "Three-way confusion matrix for classification: A measure driven view," *Inf Sci (N Y)*, vol. 507, pp. 772–794, Jan. 2020, doi: 10.1016/j.ins.2019.06.064.
- [44] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," *Technologies (Basel)*, vol. 9, no. 4, Dec. 2021, doi: 10.3390/technologies9040081.
- [45] N. Namdev and N. Tomar, "Ad Click Prediction: A Comparative Evaluation of Logistic Regression and Performance Metrics," *Int J Res Appl Sci Eng Technol*, vol. 11, no. 7, pp. 1514–1523, Jul. 2023, doi: 10.22214/ijraset.2023.54914.
- [46] A. Abdullah, H. Chaturvedi, S. Fuladi, N. J. Ravuri, D. Natesan, and M. K. Nallakaruppan, "Reliable and Efficient Model for Water Quality Prediction and Forecasting." [Online]. Available: www.ijacsa.thesai.org
- [47] D. Rodriguez, "Area under the Curve as an Alternative to Latent Growth Curve Modeling When Assessing the Effects of Predictor Variables on Repeated Measures of a Continuous Dependent Variable," *Stats (Basel)*, vol. 6, no. 2, pp. 674–688, Jun. 2023, doi: 10.3390/stats6020043.
- [48] G. Zeng, "On the confusion matrix in credit scoring and its analytical properties," *Commun Stat Theory Methods*, vol. 49, no. 9, pp. 2080–2093, May 2020, doi: 10.1080/03610926.2019.1568485.
- [49] D. Rodriguez, "Assessing Area under the Curve as an Alternative to Latent Growth Curve Modeling for Repeated Measures Zero-Inflated Poisson Data: A Simulation Study," *Stats (Basel)*, vol. 6, no. 1, pp. 354–364, Mar. 2023, doi: 10.3390/stats6010022.
- [50] K. Budholiya, S. K. Shrivastava, and V. Sharma, "An optimized XGBoost based diagnostic system for effective prediction of heart disease," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4514–4523, Jul. 2022, doi: 10.1016/j.jksuci.2020.10.013.
- [51] F. S. Nahm, "Receiver operating characteristic curve: overview and practical use for clinicians," *Korean J Anesthesiol*, vol. 75, no. 1, pp. 25–36, Feb. 2022, doi: 10.4097/kja.21209.
- [52] A. Helisa, H. Saragih, I. Budiman, F. Indriani, D. Kartini, and T. H. Saragih, "Prediction of Post-Operative Survival Expectancy in Thoracic Lung Cancer Surgery Using Extreme Learning Machine and SMOTE," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 2, pp. 239–249, 2023, doi: 10.26555/jiteki.v9i2.25973.
- [53] W. Jiang, Z. Li, J. Li, Y. Zhu, and P. Zhang, "Study on a fault identification method of the hydraulic pump based on a combination of voiceprint characteristics and extreme learning machine," *Processes*, vol. 7, no. 12, Dec. 2019, doi: 10.3390/PR7120894.

AUTHORS BIOGRAPHY



Nor Aida, a native of Banjarmasin, South Kalimantan, embarked on her academic journey after completing high school. Since 2020, she has been diligently pursuing a degree in Computer Science at Lambung Mangkurat University. Her research interests are centered on the burgeoning field of data science. To augment her skill set and acquire relevant certifications, Aida participated in the prestigious Bangkit Academy program in 2023, specializing in the Cloud Computing learning path. Her final project involved a comprehensive investigation into predicting the recurrence of differentiated thyroid cancer by conducting a comparative analysis of the Extreme Learning Machine and Hidden Markov Model algorithms, employing the SMOTE technique to enhance data balance.



Triando Hamonangan Saragih is a dedicated lecturer member at the Department of Computer Science, Lambung Mangkurat University, where he imparts knowledge in a diverse range of subjects including calculus, discrete mathematics, and other relevant disciplines. His scholarly pursuits are primarily focused on the dynamic fields of Data Science and Computer Networks. He has earned both his bachelor's and master's degrees in Computer Science from Brawijaya University, Malang, in 2016 and 2018. Respectively, Saragih's research endeavors delve into the intricacies of data science, machine learning, and their potential applications, demonstrating a commitment to advancing the frontiers of knowledge in these domains.



Dwi Kartini a lecturer with a strong basis in computer science. Having obtained both her bachelor's and master's degrees from the Faculty of Computer Science at Putra Indonesia "YPTK" Padang, Indonesia, she has dedicated her career to teaching and research in this field. As a lecturer in the Department of Computer Science,

Faculty of Mathematics and Natural Sciences, Lambung Mangkurat University in Banjarbaru, Indonesia, She imparts her knowledge to students through courses such as linear algebra, discrete mathematics, and research methodologies. Her scholarly interests converge on the practical applications of Artificial Intelligence and Data Mining, making her a valuable asset to the academic community.



Radityo Adi Nugroho a multifaceted academic and industry professional with a strong background in computer science. Having earned a bachelor's degree in Informatics from the Islamic University of Indonesia and a master's degree in Computer Science from Gadjah Mada University, he now contributes as an assistant professor at the Department of Computer Science, Lambung Mangkurat University. His research interests lie in the critical areas of software defect prediction and computer vision, where he seeks to make significant contributions to the field. Beyond academia, He's practical experience as a project manager and systems analyst in the information technology industry has equipped him with valuable skills to develop software and information systems for universities, bridging the gap between theoretical knowledge and real-world applications.



Dodon Turianto Nugrahadi is a dedicated academic contributing to the Department of Computer Science at Lambung Mangkurat University. His scholarly interests converge on the dynamic fields of Data Science and Computer Networking. Having established a strong foundation with a bachelor's degree in Informatics Engineering from UK Petra, Surabaya in 2004, He furthered his academic pursuits by obtaining a master's degree in Information Engineering from Gadjah Mada University, Yogyakarta in 2009. His current research endeavors delve into the complexities of networks, data science, the Internet of Things (IoT), and network Quality of Service (QoS), demonstrating a commitment to advancing knowledge in these critical areas.