CLASSIFICATION MODEL FOR HYPERTENSION WITH DIABETES USING GRADIENT BOOSTING AND FEATURE ENGINEERING

MONGKHON SINSIRIMONGKHON

MASTER OF ENGINEERING
IN
COMPUTER ENGINEERING

SCHOOL OF APPLIED DIGITAL TECHNOLOGY
MAE FAH LUANG UNIVERSITY
2024

CLASSIFICATION MODEL FOR HYPERTENSION WITH DIABETES USING GRADIENT BOOSTING AND FEATURE ENGINEERING

MONGKHON SINSIRIMONGKHON

THIS THESIS IS A PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING
IN
COMPUTER ENGINEERING

SCHOOL OF APPLIED DIGITAL TECHNOLOGY
MAE FAH LUANG UNIVERSITY
2024

THESIS APPROVAL

MAE FAH LUANG UNIVERSITY

FOR

MASTER OF ENGINEERING IN COMPUTER ENGINEERING

Thesis Title: Classification Model for Hypertension with Diabetes using Gradient Boosting and Feature Engineering

Author: Mongkhon Sinsirimongkhon

Examination Committee:

Associate Professor Adisorn Leelasantitham, Ph. D.	Chairperson
Associate Professor Punnarumol Temdee, Ph. D.	Member
Assistant Professor Sujitra Arwatchananukul, Ph. D.	Member
Associate Professor Nattapol Aunsri, Ph. D.	Member
Surapong Uttama, Ph. D.	Member

Advisors:

P. Temdee .....Advisor

(Associate Professor Punnarumol Temdee, Ph. D.)

Sujitra A. .....Co-Advisor

(Assistant Professor Sujitra Arwatchananukul, Ph. D.)

Dean:

.....

(Assistant Professor Nacha Chondamrongkul, Ph. D.)

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to everyone who supported me throughout my journey in pursuing my master's degree.

I am profoundly thankful to my advisor, Assoc. Prof. Dr.Punnarumol Temdee, for her unwavering support and patience and for allowing me to grow academically and personally.

I would also like to sincerely thank my co-advisor sincerely, Asst. Prof. Dr.Sujitra Arwatchananukul, for her insightful advice and continuous encouragement.

A big thank you goes to my friends, whose help and companionship made this journey easier and more enjoyable.

Mongkhon Sinsirimongkhon

Thesis Title	Classification Model for Hypertension with Diabetes Using Gradient Boosting and Feature Engineering
Author	Mongkhon Sinsirimongkhon
Degree	Master of Engineering (Computer Engineering)
Advisor	Assoc. Prof. Punnarumol Temdee, Ph. D.
Co-Advisor	Asst. Prof. Sujitra Arwatchananukul, Ph. D.

ABSTRACT

Hypertension and diabetes present significant global health challenges, impacting individual well-being and economies. Early detection and prevention are pivotal in mitigating their adverse effects. Machine learning is widely applied in various industries and has shown promise in healthcare. While machine learning has shown promise in predicting these conditions separately, limited research has focused on their co-occurrence. This study proposes a novel multiclass-classification approach to predict the coexistence of hypertension and diabetes. The methodology encompasses data collection, preprocessing, model construction, validation, and comparison. Various classifiers were employed, including Decision Tree, Support Vector Machines, Random Forests, Extra Trees, Gradient Boosting, and Long Short-Term Memory. Additionally, CTGAN was utilized to address imbalanced datasets. Results demonstrate the effectiveness of the proposed approach. Gradient Boosting emerged as the most successful among the classifiers, achieving an impressive accuracy of 92.21% and an average AUC-ROC of 96.46%. These findings underscore the potential of machine learning in accurately predicting the concurrent presence of hypertension and diabetes. This study's significance lies in its contribution to understanding and diabetes.

This study's significance lies in its contribution to understanding and predicting complex health conditions, facilitating early intervention and personalized care strategies. The outcomes suggest a promising avenue for healthcare practitioners to enhance proactive management approaches for individuals with both hypertension and diabetes

Keywords: Multiclass Classification, Machine Learning, Feature Engineering, CTGAN

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background and Problems

Noncommunicable diseases (NCDs) or chronic diseases constitute conditions not due to infection but a combination of genetic, physiological, environmental, and behavioral factors. These conditions have lasting health consequences and often require long-term treatment and care. NCDs are believed to cause approximately 41 million deaths each year, equivalent to 74% of all deaths globally [1]. They are most frequently associated with older age groups. However, evidence suggests that over 15 million of all deaths attributed to NCDs occur in people between the ages of 30 and 69 years [1]. Early detection, screening, and treatment are critical components of the response to NCDs.

Hypertension and diabetes are NCDs that are major public health concerns. Because they share several common causes and risk factors, a person with one condition is at an increased risk of developing the other. In addition, these are two of the significant risk factors for cardiovascular diseases (CVDs) [2]. Generally, patients with type 2 diabetes have a greater chance of having high blood pressure. In the US, it is estimated that 73.6% of individuals with diabetes who are aged 18 years or more tend to have hypertension [3].

Further, 50%–80% of patients with hypertension tend to have type 2 diabetes [4]. In Hong Kong, 58% of diabetic patients have high blood pressure, and 44% of hypertensive people also have diabetes [2]. The same trend is observed in Thailand, where the number of patients with both hypertension and diabetes is increasing [5]. According to the twelfth five-year National Health Development Plan (2017–2021), these diseases are categorized as national priority diseases [6].

Hypertension is a disease that occurs when the heart contracts to pump blood through the arteries to the entire body. The pressure increases when the cardiac muscles contract and decreases when they relax. However, if the muscles are relaxed but the pressure in the blood vessels does not drop below the specified threshold value of systolic blood pressure (SBP), 140 mm Hg, and that of diastolic blood pressure (DBP), 90 mm Hg. [7], the patient is considered to have hypertension. This disease could lead to complications in essential organs, including the heart, which is forced to work harder. This causes the cardiac muscles and the myocardium wall to simultaneously become thicker and weaker, eventually leading to heart failure [8]. The World Health Organization (WHO) reported that 46% of adults with hypertension are unaware of their condition [9].

Hypertension patients are typically diagnosed based on their blood pressure values. A medical professional performs the primary examination. If the blood pressure exceeds 140/90 mm Hg [7], they must be examined again to verify whether they have primary or secondary hypertension. While primary hypertension occurs naturally, secondary hypertension is either caused by other diseases or occurs as a side effect due to certain medications [8]. In addition, the patients are examined for damage to their internal organs caused by hypertension. Then, the medical professional provides the appropriate methodology for stabilizing the blood pressure through lifestyle modifications, such as maintaining an appropriate Body Mass Index (BMI) and weight, refraining from smoking and exercising regularly. Finally, the proper treatment is provided according to each patient's requirements.

Diabetes is a condition in which the body has high blood sugar levels caused when starch and sugar are consumed but cannot be absorbed by the body for use. The main reason for this is inadequate production of the hormone insulin in the pancreas [10]. Thus, the body cannot transmit sugar in the form of glucose in the bloodstream to other tissue systems to burn and convert into energy for the body to use. Another reason this occurs is tissue or organs' resistance to insulin. As a result, the amount of sugar in the body remains in the bloodstream in large quantities. If the patients are unaware that they have diabetes and do not adjust their behavior accordingly, it might lead to various complications in the future [11-13]. The WHO reported that diabetes was the direct cause of 1.5 million deaths in 2019 alone [14].

Diabetic patients are diagnosed based on their blood sugar levels [10]. In general, the glucose level of people who do not have diabetes ranges between 70–99 mg/dL before the first meal of the day or breakfast and does not exceed 140 mg/dL within 2 hours of eating. Depending on their blood sugar levels, patients can be divided into varying levels of the disease [15]. A blood sugar level above 200 mg/dL is considered diabetic regardless of whether the person has eaten or fasted before testing. If the blood sugar level before breakfast falls between 100 mg/dL and 125 mg/dL, it is assessed as abnormal or at risk. As a result, the patient must be examined and followed up every year. If the value exceeds 126 mg/dL before breakfast, it is diagnosed as diabetes. However, if the level is less than 126 mg/dL, the patient is tested by drawing blood before and 2 hours after drinking a glucose solution.

With advancements in technology related to healthcare applications, medical data has been used for many purposes, such as disease diagnosis [16-23], symptom tracking [24], lifestyle behavior adjustment [25], and disease prediction [26-27]. Advanced research methods have been widely developed for disease prediction. Recently, the subject of hypertension associated with diabetes has been gaining much interest because the two diseases often co-occur [28]. Many studies have adopted machine learning algorithms to create a classification model to predict only diabetes [29-31] and hypertension [32-35]. However, models aimed at diagnosing hypertension with diabetes are rare.

Among machine learning methods, the ensemble learning method is widely used to create models in many fields, primarily the healthcare domain. These methods aim to improve performance by aggregating the predictions of multiple estimators. This, in turn, enhances model performance, which could lead to early detection and, more importantly, prevention.

Data imbalance happens when the majority class and minority class have a significant difference and can lead to a biased model. It’s not a rare condition, and the nature of the dataset. CTGAN is a technique to create synthetic data that generates data that looks like the original dataset. This technique generates synthetic samples that can be generated for the minority classes, which balances the class distribution by ensuring that each class is treated fairly. This improved class balance allows ensemble learning

methods to make more accurate predictions on all classes, including the minority classes.

This research proposes a multiclass classification of the coexistence of hypertension and diabetes models using machine learning. Feature engineering was also applied to make raw datasets suitable for machine learning algorithms, including feature imputation to handle missing data, feature construction to transform and encode data to a more usable format for machine learning algorithms, and handling the imbalance dataset issue by using CTGAN.

1.2 Research Objectives

The objective of this study is to develop a predictive model using machine learning to detect hypertension, diabetes, and hypertension with diabetes.

1.3 Scope

This study primarily focuses on feature engineering-based approaches for achieving optimal multiclass classification of patients with diabetes, hypertension, or diabetes with hypertension. The objective is to leverage a historical dataset of patients' medical records and apply various feature engineering techniques to enhance the performance of classification models. The study aims to identify the most compelling feature engineering strategies that contribute to accurately classifying patients into the three target classes: diabetes, hypertension, and diabetes with hypertension.

1.4 Thesis Structure

The remainder of this thesis is organized as follows, and a brief content summarizing each chapter is also given.

Chapter 2: Literature Review. This chapter provides related research and similar fields of interest.

Chapter3: Feature Engineering. This chapter presents the experiment's procedure and how it was implemented in previous work.

Chapter 4: Methodology. This chapter presents the procedure for the experiment and how to implement it in the proposed study.

Chapter 5: Results and Discussion. This chapter presents current progress, including the results of traditional and ensemble learning methods.

Chapter 6: Conclusion This chapter presents the summary of the objectives of the research and discusses future research.

CHAPTER 2

LITERATURE REVIEW

2.1 Related Work

Machine learning–based methods are effective classifiers for disease prediction models, especially for hypertension and diabetes.

2.1.1 Diabetes Prediction Models

Lama et al. [29] employed RF relatively successfully to identify people with increased type 2 diabetes or pre-diabetes risk without known abnormal glucose regulation. The features included personal and clinical data, such as BMI, waist-hip ratio, age, systolic and diastolic blood pressure, and diabetes heredity. Mirzajani and Salimi [30] developed a model to diagnose diabetes by using several machine learning methods, including Artificial Neural Network (ANN), Basin Network, DCT (C5.0), and SVM. According to their study, DCT (C5.0) performed the best among the algorithms used. Sonar and Jayamalini [31] proposed a model to predict diabetic risk levels, with DCT as the best classifier compared to ANN, Naïve Bayes, and SVM. Rahman et al. [32] using the Pima Indians Diabetes Database. Significant features were extracted from the dataset using Boruta algorithms. Features included glucose, BMI, insulin, blood pressure, and age. Hyperparameter optimization was also performed using Grid search; the best model is Conv-LSTM, which achieved the highest accuracy of 97.26%. Umair et al. [33] used the Pima Indian Diabetes Dataset is used, statistical algorithms, and machine learning algorithms, and LSTM able to achieve the highest accuracy of 87.26%, Swapna et al. [34]. They used ECG signal as input, and they used CNN, CNN-LSTM, and CNN-LSTM with SVM, and CNN-LSTM with SVM was able to achieve the highest accuracy of 95.7%

2.1.2 Hypertension Prediction Models

Similarly, several studies have attempted to predict hypertension as well. Nasir et al. [35] predicted blood pressure-related disorders and cardiovascular diseases based on a blood pressure dataset obtained from Kaggle. This dataset included personal data as well as clinical data, including blood pressure abnormalities, gender, BMI, and age. Using four types of machine learning algorithms, including Random Forest (RF), CatBoost, Support Vector Machine (SVM), and K-nearest neighbor (KNN), the authors observed that CatBoost and RF outperformed the other algorithms. AlKaabi et al. [36] constructed and compared models for identifying patients with a high risk of hypertension, with RF outperforming other algorithms. The model included various features associated with personal and behavioral data, such as age, gender, education level, employment, tobacco use, physical activity, consumption of fruits and vegetables, abdominal obesity, history of diabetes, history of high cholesterol, and mother's history of high blood pressure. Jain et al. [37] focused on predicting the likelihood of abnormality in blood pressure. Fifteen features representing personal data, clinical data, and behavioral data were taken into consideration, including kidney disease, adrenal, and thyroid disorder, level of hemoglobin, genetic pedigree coefficient, age, BMI, sex, pregnancy, smoking, physical activity, input salt content in diet, alcohol consumption per day, input level of stress, and blood pressure abnormality. The authors applied various classifiers, such as Naïve Bayes, SVM, RF, Gradient Boosting (GB), and Logistic Regression (LR). GB and RF showed the best results compared to other algorithms. Yen et al. [38] use photoplethysmography signals to classify hypertension into no hypertension, prehypertension, stage I, and stage II. PPG signals were used to train deep residual network convolutional neural network (ResNetCNN) and bidirectional long short-term memory (BILSTM). Data consisted of 2100 data points, and the model from BILSTM demonstrated an optimal classification accuracy of 76%. Choi et al. [39] use a Korean National Health Insurance Service database dataset. Patients with hypertensive disease who received more than five health screenings between 2002 and 2011 were selected. Four models were established on both logistic and deep learning methods. All datasets' deep learning basis models showed higher receiver operating characteristics area under the curve (AUC) values

than logistic regression models. The deep learning model showed the highest AUC value of 0.954.

2.1.3 Multiclass Classification Model

Recently, a limited number of studies have attempted to predict both hypertension and diabetes. For instance, Fitriyani et al. [40] proposed a model for the early prediction of type 2 diabetes and hypertension by using ensemble machine learning–based methods. However, two separate models were developed to predict hypertension and diabetes separately.

Currently, limited studies focus on the multiclass classification of diabetes with hypertension. Oliveira et al. [41] proposed a comparative study of machine learning techniques for multiclass classification, including Dengue, Chikungunya, and others. They used clinical and socio-demographic data from patients. A feature selection technique was also applied in the process. Gradient boosting outperforms other metrics algorithms.

Khan et al. [42] optimized deep learning methods for multiple stomach disease classification. The preprocessing fuses filtering images with Ant Colony Optimization, deep transfer learning-based features extraction, optimizing deeply extracted features using nature-inspired algorithms, and a Multi-layered Perceptron Neural Network for classification.

2.1.4 Model with Data Augmentation CTGAN

Habibi et al. [43] use CTGAN to handle imbalanced tabular datasets. Their study's objective is to apply the CTGAN model to tabular data modeling to overcome the imbalance issue commonly found in IoT botnet datasets. After augmenting the data using CTGAN, they proposed an MLP model that achieved an accuracy of 98.93%.

Jia et al. [44] address the problem of insufficient data and the imbalance dataset between the number of normal and failure data. They used RCTGAN with four classifiers multilayer perceptron, support vector machine, decision tree, random forest, enhanced version of CTGAN to improve its performance; the results show that the data synthesized by the RCTGAN can further improve the accuracy of classifier MLP, which is able to achieve 80.5% accuracy.

Wang et al. [45] they study present a traffic sample synthesizing model named Conditional Tabular Traffic Generative Adversarial Network (CTTGAN), which uses CTGAN to expand the small category traffic samples and balance the dataset.

The experimental results show that the recognition rate of the expanded samples is more than 0.99 in MLP, KNN, and SVM.

Majeed et al. [46] propose a data augmentation scheme called conditional generative adversarial network minority-class-augmented oversampling scheme (CTGAN-MOS) for solving class imbalance problems by adding synthetic samples only to the minority class. The result proves the CTGAN-MOS has yielded an accuracy value of 100%.

In contrast, this study proposes a single disease prediction model for both hypertension and diabetes. More specifically, a multiclass classification model is developed to predict the diabetes group, hypertension group, and hypertension with diabetes group. The literature review reveals that several types of machine learning methods are widely accepted for disease prediction models, especially ensemble machine learning–based methods. Thus, this study mainly focuses on constructing the multi-class classification model using several potential machine learning methods, including RF, GB, ET, SVM, DCT, and LSTM, and their associated feature engineering. This study employs data from Phaya Mengrai Hospital and Theong Hospital, Chiang Rai, Thailand, including personal, clinical, and behavioral data. The prediction results of each model, with and without feature engineering and CTGAN, were compared to evaluate the prediction performance.

2.2 Computational Theory

2.2.1 Data Cleansing

Data cleansing or cleaning is a subprocess in data preprocessing that focuses on identifying and correcting or removing errors, inconsistencies, and inaccuracies in the data. The object is to improve the quality and reliability of the data before using it to train machine learning models.

Interquartile Range (IQR) [47] technique is a commonly used statistical method for detecting and handling outliers in a dataset. IQR is the result of subtraction between the third (Q3) and first quartile (Q1) of a distribution, as shown in Equation (1).

The upper and lower bound values are obtained from Equations (2) and (3). An outlier can be detected if the data point's value is higher than the upper value or less than the lower bound. Values that fail within the upper bound and lower bound are included in the dataset.

$$IQR=Q3-Q1 \quad (1)$$

$$\text{Upper bound}=Q3+1.5*IQR \quad (2)$$

$$\text{Lower bound}=Q1-1.5*IQR \quad (3)$$

2.2.2 Feature Engineering

Feature engineering [48] is the process of transforming or creating new features from existing raw data to improve the performance and interpretability of machine learning models because relying on machine learning algorithms to extract underlying patterns from poorly formatted data might affect the effectiveness of the machine learning model.

2.2.2.1 Feature Improvement

Feature improvement [48] is a process under feature engineering; the purpose is to make existing features more usable by applied mathematical transformations, such as imputing missing data by using arbitrary values or inferring them from the other column.

Arbitrary imputation [48] which will insert arbitrary value to the missing field to indicate missingness as a separate category, which might have meant or represented a distinct category,

The most frequent imputation [48] will use the most repeat or mode value. This preserves the distribution and minimizes information loss since the imputed values are based on the observed values that already exist in the dataset. It is useful when the missingness is assumed to be random or when no specific pattern is related to the missing values.

End-tail imputation [48] is a technique for imputing missing values for continuous features by replacing them with extreme values from the tails of the distribution. It indicates when missing values are assumed to be non-random and may carry specific meaning or information.

2.2.2.2 Feature Construction

Feature construction [48] is a process under feature engineering; it creates new interpretable features from existing interpretable features.

Ordinal encoding [48] or label encoding is a technique for transforming categorical variables into numerical representations. It assigns a unique numerical label to each category of the variable to preserve the ordinal relationship.

One-hot encoding [48], which represents categorical variables as binary features, where the presence or absence of a category is indicated by 1 or 0, helps to remove implicit ordering and maintain categorical information.

Yeo-Johnson transformation [49] transforms continuous features to achieve a more Gaussian-like distribution. It can handle zero and negative values, which can be beneficial for improving the performance of certain statistical analyses or machine learning models that assume normality.

The formula for Yeo-Johnson is shown below in formula (4).

$$h_\lambda(x) = \begin{cases} \dfrac{(1+x)^\lambda - 1}{\lambda} & if\ \lambda \neq 0\ and\ x \geq 0 \\ \log(1+x) & if\ \lambda \neq 0\ and\ x \geq 0 \\ -\dfrac{\left((1-x)^{2-\lambda} - 1\right)}{2-\lambda} & if\ \lambda \neq 2\ and\ x < 0 \\ -\log(1-x) & if\ \lambda \neq 2\ and\ x < 0 \end{cases} \quad (4)$$

Where X is the observed feature, which can be a 0 or negative value, λ is the real value as a parameter to tune the distribution. $\lambda < 1$ will transform right skewed toward the symmetry, and for $\lambda > 1$ will transform left skewed toward the symmetry.
Normalization [49] is a process used in data preprocessing to scale features to a specific range, often [0, 1], to ensure that all features contribute equally to the model. The formula is shown below in formula (5).

$$X' = \frac{X - Xmin}{Xmax - Xmin} \quad (5)$$

2.2.3 Model Construction and Validation

Model construction and validation are steps in the machine learning process to ensure that the developed model is reliable, accurate, and capable of making meaningful predictions.

2.2.3.1 Model Construction

Model construction is the process of building a predictive or decision-making model using a given dataset. In this study, the dataset was split into two parts: the training and test datasets. The training dataset was used to train the model. The test data was treated as unseen data that would be used to test model performance.

Classifiers are algorithms or models that assign a category or label to input data based on its features. They are a fundamental component of supervised learning, where the models learn from labeled training examples to make predictions or decisions on unseen data.

Traditional supervised learning methods refer to the fundamental algorithm that has been widely used in machine learning for many years.

Decision Tree Classifiers [50] are among the oldest and most used supervised learning methods. They create a tree-like model where each internal node represents a test on a feature, each branch represents the outcome, and each leaf node represents a class label. They can interpret both categorical and numerical data.

Equation (6) represents the Gini impurity equation, a commonly used criterion for evaluating the data's impurity and guiding the splitting process in decision tree algorithms. The Gini impurity provides a measure of how impure or heterogeneous the class distribution is within a given node.

$$Gini(p) = 1 - \sum_{i=1}^{n} (p_i)^2 \qquad (6)$$

Where Gini(p) is the Gini impurity for a given node, n is the number of classes, and pi is the probability of an instance belonging to class i. The Gini impurity ranges from 0 to 1, where 0 represents perfect purity (all instances belong to the same class), and 1 indicates maximum impurity (instances are evenly distributed across classes).

Support Vector Machines, or SVM [50], are also a traditional supervised learning method. SVM classifiers aim to find the optimal hyperplane separating different feature space classes. They are particularly effective when the data is separable by a clear margin. SVM can also handle linear and non-linear classification tasks using different kernel functions. The equation for SVM can be expressed as equation (7).

$$f(x) = sign(\sum_{i=1}^{n_s} y_i \, \alpha_i \, K(x, x_i) + b) \qquad (7)$$

Where f(x) represents the predicted class label for the input x, the sign is the sign function that assigns the positive class or negative class based on the sign of its arguments. n_s is the number of support vectors, y_i represents the class label of the i-th support vector, α_i denotes the corresponding Lagrange multiplier or weight associated with the i-th support vector, K (x, x_i) represents the kernel function that computes the similarity between the input instance x and the i-th support vector x_i, and b is the bias term.

Ensemble learning methods [51] in supervised machine learning involve combining multiple individual models to make more accurate predictions or decisions. By leveraging multiple models' diversity and collective intelligence, ensemble methods can often outperform a single model.

Ensemble learning bagging or bootstrap aggregating [51] is one of the ensemble learning method categories. Bagging involves randomly sampling the training data with replacement to create multiple subsets, and each subset is the same size as the original training set. A separate model is trained on each sample independently. The models are trained in parallel, which allows for efficient computation. For classification tasks, the predictions of individual models are combined using majority voting. This study uses Random Forest and Extra Trees as a candidate from ensemble learning bagging methods. Equation (8) shows Out-of-Bag (OOB). The OOB is calculated by evaluating the predictions of each tree on the instances not included in its corresponding bootstrap sample. This provides a way to estimate the model's accuracy without needing a separate validation set.

$$OOB\ Error = 1 \sum_{i=1}^{n} L(y_i, \hat{Y}_i^{OOB}) \qquad (8)$$

Where OOB Error represents the estimated error rate of the model using the out-of-bag instances, n is the total number of instances, y_i represents the true class label of instance I, Yˆ_i^OOB is the predicted class label of instance i based on the votes. L is the loss function or error metric used to compare the true class with the predicted class.

Random Forest [51] is a popular candidate for the bagging ensemble method. It can combine multiple decision tree trained on different subsets of the data and features. The final prediction is obtained by aggregating the predictions of all individual trees. Figure 2.1 illustrates the functioning of this ensemble method.

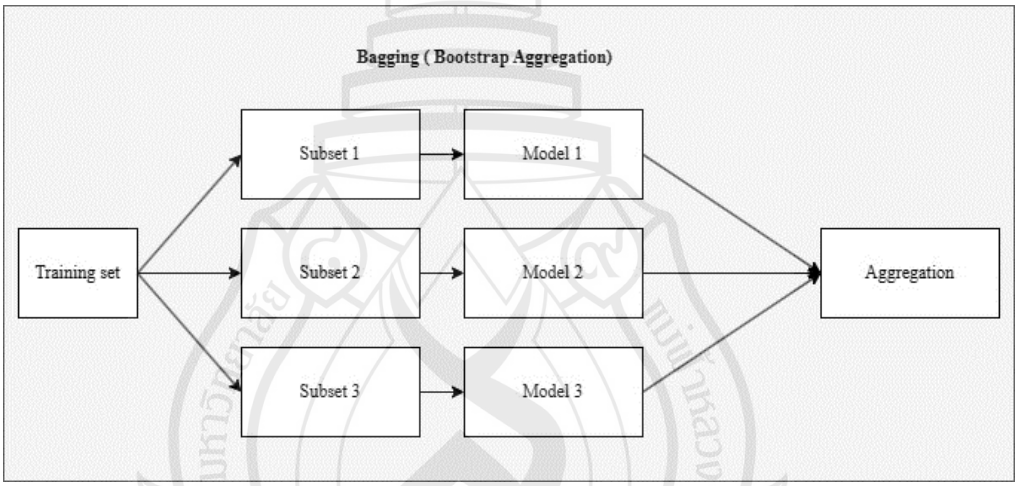


Figure 2.1 Bagging methods

Extra Trees [51], short for Extremely Randomized Trees, is an ensemble learning method that extends the concept of random forests. Like random forests, it builds a collection of decision tree, where each tree is trained on a random subset of the training data. In Extra Trees, rather than considering all features to determine a Random Forest to find the best split at each decision tree node, a random subset of features is selected, making the algorithm even more random and less influenced by individual features.

The ensemble learning boosting [51] method is one of the ensemble learning method categories. The boosting method trains a sequence of models iteratively.

Each model is trained to correct the mistakes made by previous models. The model is trained sequentially. The predictions of individual models are combined by giving more weight to the modes with better performance. The final prediction is obtained by aggregating the predations of all models and assigning higher weights to a more accurate model. This study used Gradient Boosting, which is under this category. Figure 2.2 illustrates the functioning of this ensemble method.

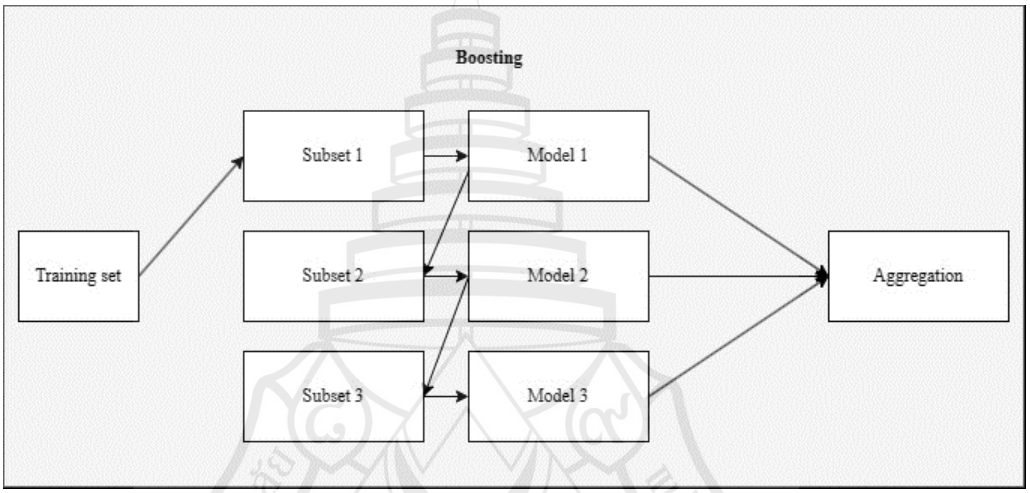


Figure 2.2 Boosting methods

Gradient descent is used as the optimization algorithm to minimize the loss function and iteratively improve the model's predictions. Equation (9) shows the formula gradient descent in the context of Gradient Boosting.

$$\theta_{t+1} = \theta_t - \alpha \nabla \mathrm{L}(y, \widehat{Yt}) \qquad (9)$$

Where θ_(t+1) represents the updated parameters at iteration t + 1, θ_t represents the current parameters at iteration t, α denotes the learning rates, L represents the loss function between true target and predicted values at iteration t, ∇L represents the gradient of the loss function with respect to the predicted values.

Gradient Boosting [51] is a popular boosting algorithm. It builds an ensemble of models by minimizing a loss function gradient and updating the model iteratively to improve the predictions.

Deep learning is also known as artificial neural networks (ANNs) [51]. Deep learning is a field of machine learning that focuses on training with multiple layers to learn and make predictions or decisions from large amounts of data. The deep learning approach has shown promising results in various domains such as computer vision, speech recognition, anomaly detection, etc.

Recurrent Neural Networks [52] (RNNs) are a type of artificial neural network designed for processing sequential data. They have feedback connections that allow information to persist and be shared across different steps or time points in the sequence. The equations for RNN cells can be expressed as equations (10) and (11).

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \quad (10)$$

$$y_t = softmax(W_{hy}h_t + b_y) \quad (11)$$

Where h_t represents the hidden state at time step t, x_t represents the input a time step t, W_hh denotes the weight matrix for the hidden state at the previous time step. W_xh denotes the weight matrix for the current input. b_h represents the bias term for the hidden state. W_hy represents the weight matrix for the output layer. b_y represents the bias term for the output layer. σ represents the activation function, and SoftMax is an activation function typically used for multi-class classification.

Long Short-Term Memory [52] (LSTM) is a type of RNN architecture that addresses the vanishing gradient problem, which can hinder the learning process in RNNs. LSTM networks are designed to capture long-term dependencies and information over extended time intervals by incorporating memory cells and gating mechanisms that regulate the flow of information within the network.

LSTM networks consist of memory cells that can store and update information over time, input gates that regulate the flow of new information into the memory cells, forget gates that control the retention or removal of information, and output gates that determine the network's output based on the memory cells' content. This design allows LSTMS to model and process sequential data effectively, making them popular in applications such as speech recognition and time series prediction.

Equations for LSTM can be expressed as follows: equation (12), equation (13), equation (14), equation (15), and equation (16).

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \quad (12)$$

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \quad (13)$$

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \quad (14)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{ch}\, h_{t-1} + W_{cx}x_t + b_c) \quad (15)$$

$$h_t = o_t \odot \tanh(c_t) \quad (16)$$

Where h_t represents the hidden state at time step t, c_t represents the cell state at time step t, x_t represents the input at time step t. f_t denotes the forget gate determining how much of the previous cell state to forget. i_t denotes the input gate determining how much new candidate values can be incorporated into the cell state. o_t denotes the output gate that determines how much of the cell state should be, σ represents the sigmoid activation function, ⊙ denotes element-wise multiplication.

The LSTM-based model can be effective for anomaly detection in time series data, including applications such as detecting anomalies in healthcare data like diabetes or hypertension because they can capture temporal dependencies and patterns in sequential data.

Data augmentation is a technique in machine learning that deals with limited datasets by creating or generating more data, which is known as synthetic data. [53] Generative Adversarial Network (GAN) is a deep learning architecture. It consists of 2 neural networks, the discriminator and the generator; the generator generates new data from a given train dataset, while the discriminator predicts whether the newly generated data belongs to the real dataset. GAN will generate newer and better versions of fake data until the discriminator no longer distinguishes fake from real data. In machine learning, GAN can increase training data size by generating more data from existing data, which can introduce more variety to the model.

Conditional Tabular Generative Adversarial Network [54] is an effective tool for synthetic data generation that is tailor-made for tabular data when columns can have various types of distribution, such as continuous and categorical data. CTGAN overcomes challenges in tabular data, such as handling imbalanced data.

Traditional GANS struggle with categorical columns because they treat all columns as continuous features. CTGAN uses Mode-specific normalization for continuous features, which normalizes data based on the observed distribution.

CTGAN also uses a training-by-sampling approach, sampling data from the training set with a higher probability for minority classes. The generator is trained to produce all categories.

CTGAN uses a conditional vector during training to ensure that the generator respects the types and dependencies between columns. The conditional vector guides the generator to produce data that matches the specified conditions, helping it learn the complex dependencies between columns.

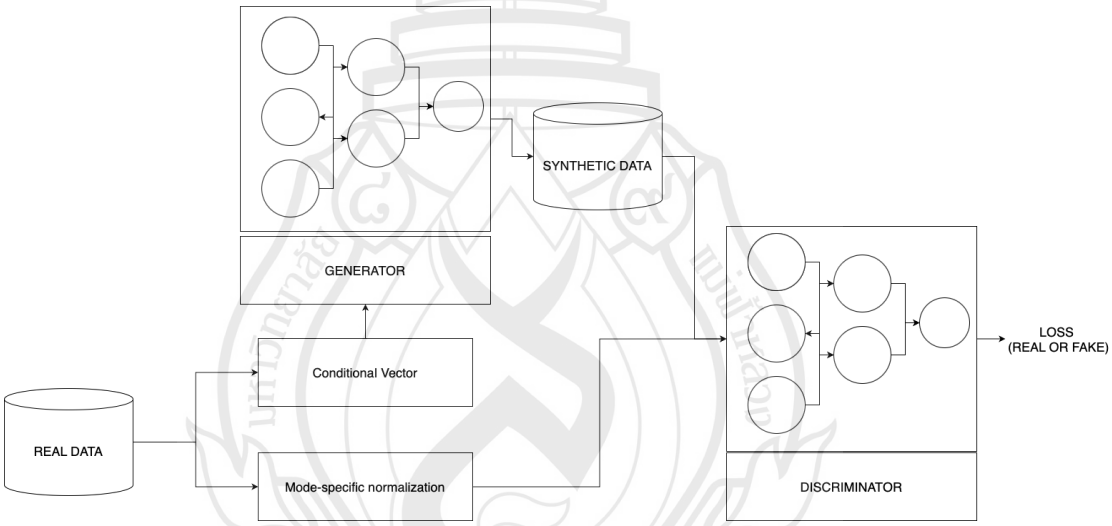


Figure 2.3 CTGAN process

According to Figure 2.3, CTGAN starts by sampling a real data point and extracting a conditional vector, a one-hot encoded category mixed with noise.

The conditional vector and random noise are passed to the generator to produce synthetic data. Send both real data and synthetic data to the discriminator to determine if the data is real or fake. Then, the loss value for both the generator and discriminator. Discriminator loss determines how well it can differentiate real from synthetic data, and generator loss measures how well the generator can trick

the discriminator. Finally, update the parameters of both networks using gradient descent based on their respective loss value.

The data validity score indicates that all the synthetic data generated by the CTGAN adheres to the expected data types. For instance, if a column of the original dataset is supposed to contain integers, the synthetic data columns will also need to contain integers.

The data structure score suggests that the synthetic data replicates the overall structure of the data, the number of columns, and their types.

Column shapes score evaluates how well the synthetic data matches the statistical distribution of each individual column in the original dataset.

Column pair trends score: the score shows the relationships between pairs of columns in the synthetic data compared to the original data. It looks at how well the correlations and interactions between columns are preserved.

2.2.3.2 Model Validation

Model validation is a crucial step in machine learning that involves assessing a trained model's performance and generalization ability on unseen data. It helps to estimate how well the model will perform when applied to real-world scenarios or new data points.

Cross-validation is a widely used technique in machine learning for model evaluation and selection. It helps assess a model's performance and generalization ability by estimating how it will perform on unseen data. Instead of relying on a single train-test split, cross-validation provides a more robust evaluation by dividing the data into multiple subsets and performing multiple train-test splits.

In data splitting, the available labeled datasets are divided into k equally sized subsets, called folds. The typical value for k is 5 or 10, but it can vary depending on the size of the dataset and the computational resources available. In iterative training and evaluation, one fold is used as the testing set for each iteration, and the remaining folds are combined to form the training set. The model is trained on the training set and evaluated on the testing set. Evaluation metrics such as accuracy, precision, recall, and F1-score are computed for each iteration based on the model's predictions on the testing set. The performance metrics obtained from each iteration are averaged to obtain

an overall performance estimate for the model. This aggregated metric serves as an indicator of the model's performance ability.

Cross-validation for hyperparameter tuning: during the hyperparameter tuning process, cross-validation is used to estimate the performance of each hyperparameter combination. The performance metrics obtained are compared after evaluating the model using cross-validation for each hyperparameter combination. The combination that results in the best performance according to the evaluation metric of interest is selected with the optimal set of hyperparameters. Once the optimal hyperparameters are obtained, the model is trained using these settings on the entire training set. The final model is then evaluated on separate testing to estimate its performance on unseen data. Figure 2.3 illustrates the functioning of k-fold validation.

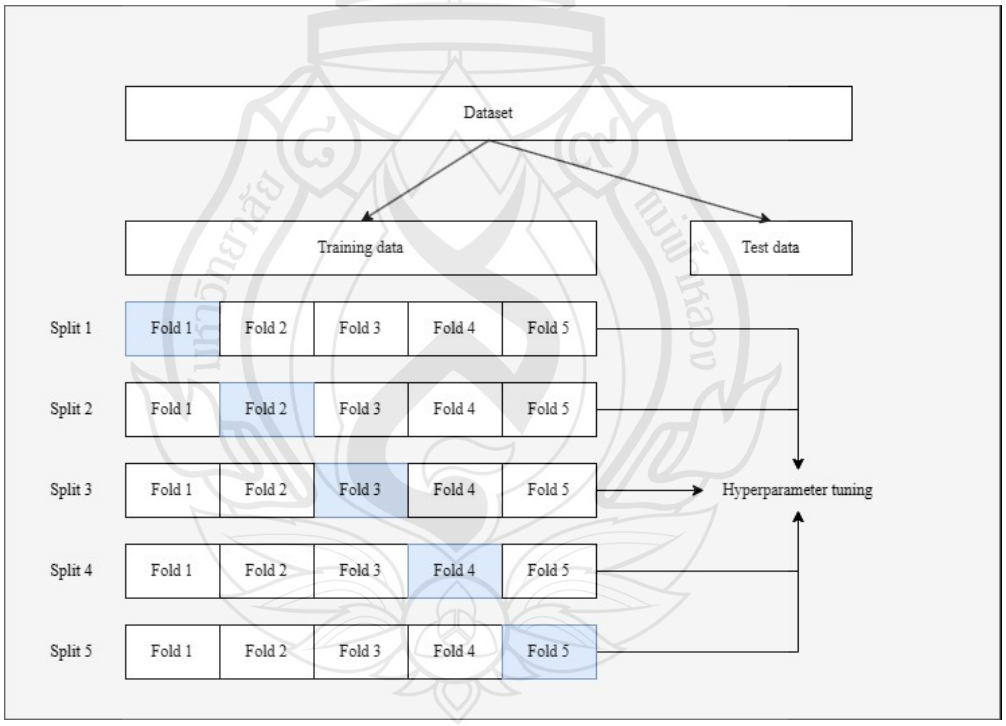


Figure 2.4 Demonstration of K-fold Cross-validation

2.2.4 Evaluation Metrics

In general, the model predictions have four different outcomes: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP and TN

indicate correct predictions. FP refers to data points classified as negative when they are negative, and FN to points classified as negative when they are positive. This study focused on accuracy and Area under the ROC curve score.

Accuracy is a metric that shows how successfully the model can make predictions. It is defined as shown in Equation (16)

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad (16)$$

Precision is the fraction of correctly predicted positive to the total predicted positive observations. The equation is shown in Equation (17).

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad (17)$$

Recall is the ratio of correctly predicted positives to all observations. It measures the model's ability to correctly identify true positives. The ratio is in Equation (18).

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad (18)$$

The F1 score is the harmonic mean of precision and recall. The equation is shown in Equation (19).

$$\text{F1 score} = 2 \text{ x } \frac{\text{Precision x Recall}}{\text{Precision+Recall}} \qquad (19)$$

Receiver Operating Characteristic (ROC) curve is a graph that shows how well the model can separate the classes. This curve plots 2 parameters: the true positive rate and the false positive rate. It is defined as shown in Equation (20) and (21) .

$$\text{True positive rate} = \frac{\text{TP}}{TP+FN} \qquad (20)$$

$$\text{False positive rate} = \frac{FP}{TN+FP} \qquad (21)$$

Area under the ROC is a metric that can represent a classifier's ability to distinguish between 2 classes. Its value ranges from 0 to 1. In the case of 100% wrong predictions, the AUC is 0; in perfectly correct predictions, the AUC ROC score is 1.

The area under the ROC metric was meant for binary classification, so the One-vs-Rest approach (OvR) must be used. The OvR method makes the binary classification method suitable for multiclass classification. This is done by splitting the multiclass dataset into multiple binary classifications, and the average values of the AUC-ROC of the classes were used to represent prediction performance. Table 2.1 will show how to make the AUR ROC score suitable for the multiclass classification problem.

Table 2.1 AUC ROC for multiclass classification

Classification Group	Positive Class	Negative Class
Binary Classification 1	Hypertension (Class 1)	Diabetes (Class 2) Hypertension with Diabetes (Class 3)
Binary Classification 2	Diabetes (Class 2)	Hypertension (Class1) Hypertension with Diabetes (Class 3)
Binary Classification 3	Hypertension with Diabetes (Class 3)	Hypertension (Class 1) Diabetes (Class 2)

CHAPTER 3

FEATURE ENGINEERING

3.1 Research Overview

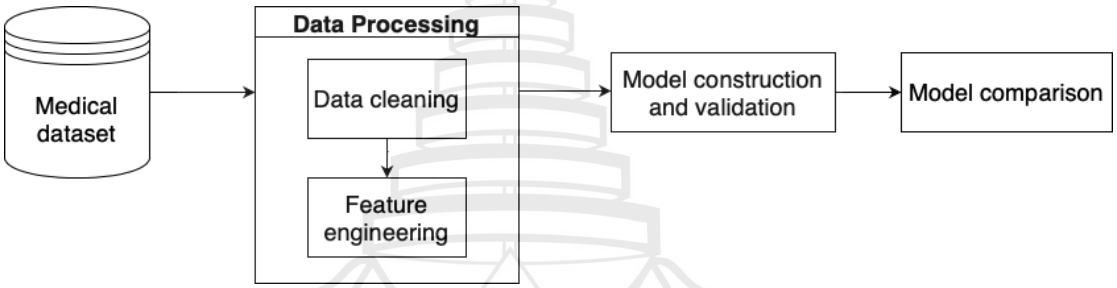


Figure 3.1 Diagram shows all the methodologies used in the process

The previous work objective was to develop a multi-class classification model for diabetes, hypertension, and diabetes with hypertension by using machine learning combined with feature engineering tactics. This section will present all the processes. Figure 3.1 shows all the processes that will be used in this study. Starting with data collection, which is gathering data for this study, data preprocessing in this step will include data cleaning, which will handle the outlier within this dataset, and feature engineering techniques for manipulating the dataset to improve the performance of machine learning models. Model construction and validation work on constructing and validating the model, and lastly, model comparison by using score metrics to indicate which model is the best.

3.2 Data Collection

In this study, historical data from patients aged between 27-102 years from 2016 until 2021 were obtained from a local hospital, Phaya Menrai Hospital, Chiang Rai, Thailand. The dataset contained 17,707 samples, and all samples represented a unique patient; there were 12,210 samples of hypertension (69%), 4,267 samples of diabetes (24%), and 1,230 samples of both hypertension and diabetes (7%). There were 28 features. The sample of the dataset is provided in Table 3.1. This study received ethical approval from the Mae Fah Luang University Ethics Committee.

Table 3.1 All features of the dataset and its type

Data Type	Features
Ordinal	Smoking behavior
	Drinking behavior
Nominal	Gender
	Family History
	Urine Albumin
	Blood–Brain Barrier
Continuous	Age
	Body Weight
	Height
	Body Mass Index
	Waist
	Respiratory Rate
	Blood Pressure Systolic
	Blood Pressure Diastolic
	Village
	Sub-district

Table 3.1 (continued)

Data Type	Features
Continuous	District
	Province
	Fasting Blood Sugar
	Total Cholesterol
	Triglyceride
	High-Density Lipoprotein
	Low-Density Lipoprotein
	Creatine
	Glomerular Filtration Rate
	Uric Acid
	Potassium
	Blood Urea Nitrogen

3.3 Data Preprocessing

Data processing is a machine learning process involving transforming raw data into a format suitable for training machine learning models. This study includes two sub-processes within this process: data cleansing or data cleaning and feature engineering. Feature engineering in this study included feature improvement, making existing features more usable by applied mathematical transformations. also, treat datasets differently depending on their characteristic; in this study able to group them into three groups, which are ordinal, nominal, and continuous. All the processes for all groups will be explained within this section.

3.3.1 Data Cleansing

This study uses the Interquartile Range (IQR) technique, a commonly used statistical method for detecting and handling outliers in a dataset.

Figures 3.2 and 3.3 show the results before and after applying the IQR method. Figure 3.2 shows a dataset of a certain feature. In this example, use the height feature. The Maximum height is 160,000 cm, which is impossible, and this kind of dataset may create a biased analysis model.

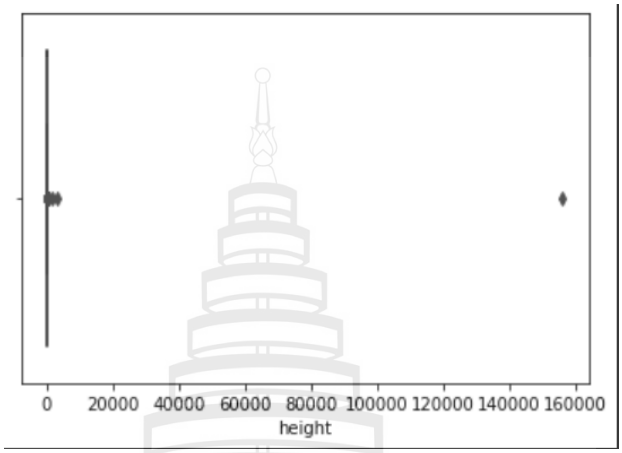


Figure 3.2 Showing the original dataset of the height feature

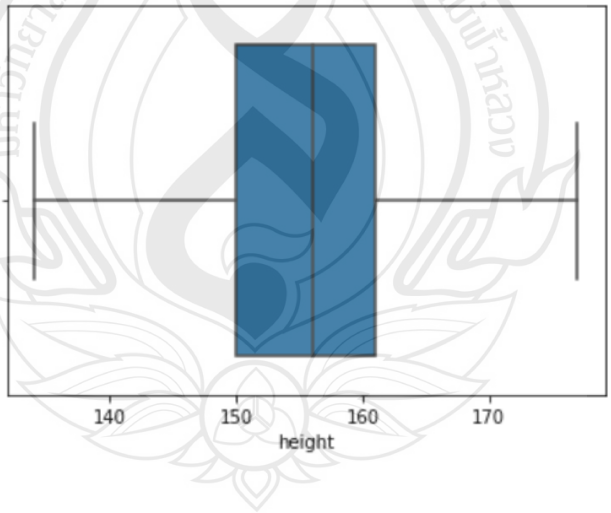


Figure 3.3 shows the dataset after applying IQR

3.3.2 Feature Engineering for Ordinal Features

In this study, features under ordinal features are shown in Table 3.2 with the feature engineering methods used.

Table 3.2 Ordinal features and feature engineering methods

Data Type	Methods	Features
Ordinal	Arbitrary Value Imputation	Smoking behavior
	Ordinal Encoding	Drinking behavior

3.3.2.1 Feature Improvement for Ordinal Features

In this study used a feature improvement technique called arbitrary value imputation. This study inserts arbitrary values into the missing field to indicate missingness as a separate category that might have meant or represented a distinct category. This study uses Feature construction for this feature.

3.3.2.2 Feature Construction for Ordinal Features

Ordinal encoding or label encoding is used to transform categorical variables into a numerical representation. It assigns a unique numerical label to each category of the variable to preserve the ordinal relationship. The image is shown in Figure 3.4.

Categories that are not seen during the encoder's training or fitting phase will be assigned the value 5 to handle new or unseen categories during the transformation of new data.

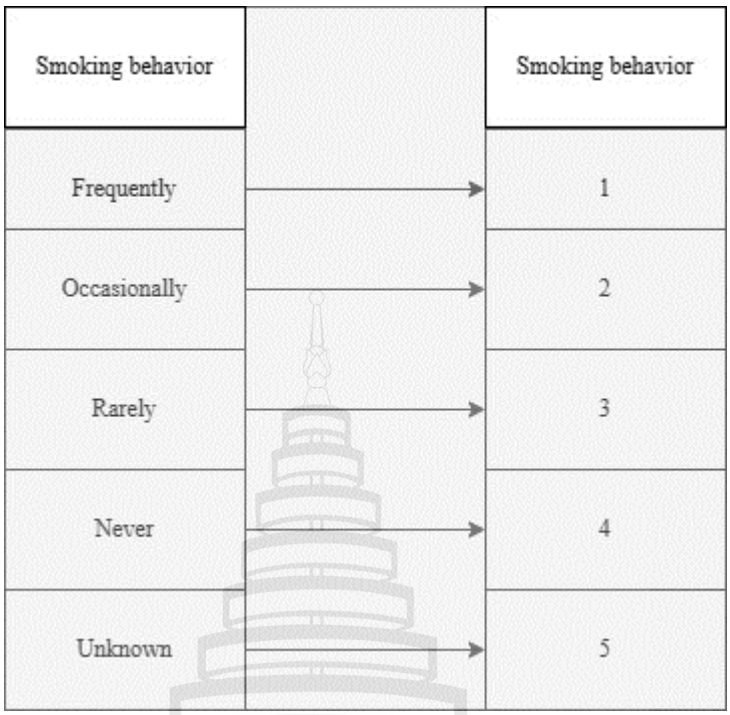


Figure 3.4 Example demonstration of ordinal encoding

3.3.3 Feature Engineering for Nominal Features

Features that are shown in Table 3.3 with feature engineering methods are used.

Table 3.3 Nominal features and feature engineering methods

Data Type	Methods	Features
Nominal	Most-Frequent imputation	Gender
	One-hot encoding	Family History
		Urine Albumin
		Blood–Brain Barrier

3.3.3.1 Feature Improvement for Nominal Features

This study uses a feature improvement technique called most-frequent value imputation by imputing the most repeated value. Doing this preserves the distribution and minimizes information loss since the imputed values are based on the observed values that already exist in the dataset and are useful when the missingness is assumed to be random or when there is no specific pattern related to the missing values.

3.3.3.2 Feature Construction for Nominal Features

For feature construction, one-hot encoding is used, which represents categorical variables as binary features, where the presence or absence of a category is indicated by 1 or 0. It helps to remove implicit ordering and maintain categorical information. Figure 3.5 illustrates the functioning of one-hot encoding.

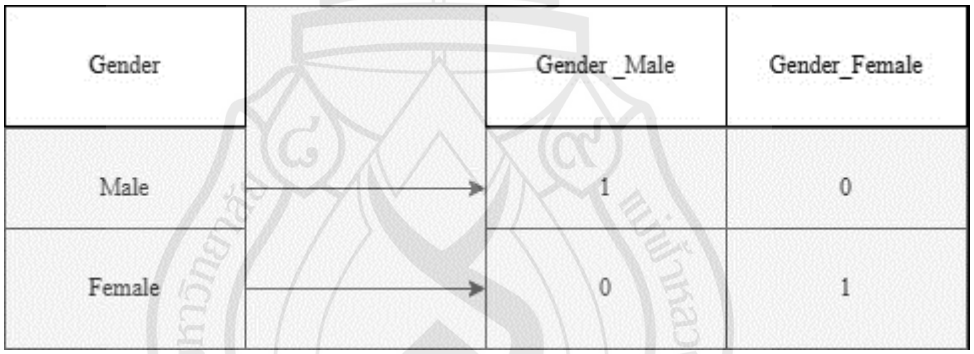


Figure 3.5 Example demonstration of one-hot encoding

3.3.4 Feature Engineering for Continuous Features

The continuous feature represents numerical variables with a wide range of possible values. Table 3.4 presents a list of continuous features along with methods.

Table 3.4 Continuous features and feature engineering methods

Data Type	Methods	Features
Continuous	Normalization	Age
	Yeo-Johnson transformation	Body Weight
	End-tail imputation	Height
		Body Mass Index
		Waist
		Respiratory Rate
		Blood Pressure Systolic
		Blood Pressure Diastolic
		Village
		Sub-district
		District
		Province
		Fasting Blood Sugar
		Total Cholesterol
		Triglyceride
		High-Density Lipoprotein
		Low-Density Lipoprotein
		Creatine
		Glomerular Filtration Rate
		Uric Acid
		Potassium
		Blood Urea Nitrogen

3.3.4.1 Feature Construction for Continuous Features

This study used normalization scaling to scale features to a range of 0 to 1 to ensure that one feature doesn't dominate another due to its scales and apply the Yeo-Johnson transformation to transform continuous features to achieve a more Gaussian-

like distribution. This study used Yeo-Johnson because this dataset has zero and negative values. The Yeo-Johnson transformation can handle zero and negative values, which can be beneficial for improving the performance of certain statistical analyses or machine learning models that assume normality.

3.3.4.2 Feature Improvement for Continuous Features

This study also uses end-tail imputation to impute missing values for continuous features by replacing them with extreme values from the tails of the distribution. This method indicates when missing values are assumed to be non-random and may carry specific meaning or information.

Figure 3.6 shows the data distribution before and after the Yeo-Johnson transformation, and it shows that the data distribution became more Gaussian-like after this technique was applied.

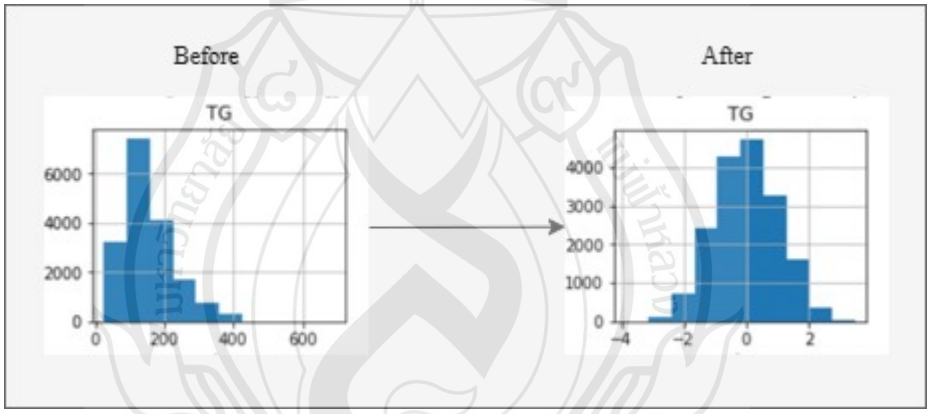


Figure 3.6 Shows before and after Yeo-Johnson transformation

3.3.5 Data Augmentation

This study used a Conditional Tabular Generative Adversarial Network to generate more sample data for the minority class by using the original dataset. Using CTGAN can generate synthetic data with high-fidelity detail and generate more samples to reduce the imbalance issue by generating the minority class to be exactly the same amount as the majority class. This can be beneficial for the model to be able to have equal classes and reduce bias.

3.4 Model Construction and Validation

Model construction and validation are steps in the machine learning process to ensure that the developed model is reliable, accurate, and capable of making meaningful predictions; figure 3.7 shows the overview of the model construction process, and model validation will be explained later in this section.

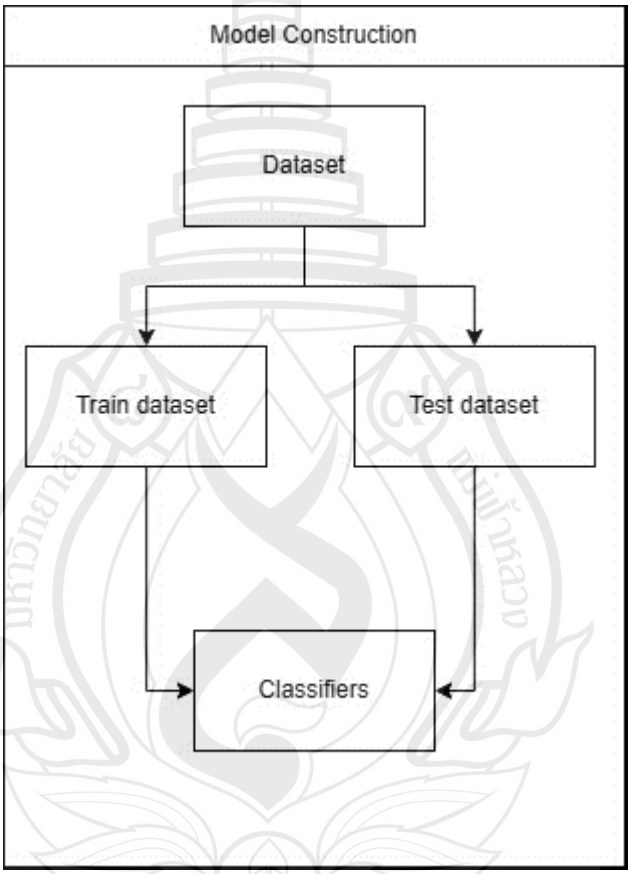


Figure 3.7 Model construction process overview

3.4.1 Model Construction

In this study, multi-class predictions were performed. This study uses 3 types of classifiers: traditional supervised learning methods, ensemble supervised learning

methods and deep learning. Classifiers that are under traditional supervised learning methods categories are decision tree (DCT) and support vector machines (SVM), and classifiers that are under ensemble supervised learning methods are Random Forest (RF), Gradient Boosting (GB), and Extra Trees (ET). For deep learning, using LSTM.

3.4.2 Model Validation

This study process of model validation is shown in Figure 3.8.

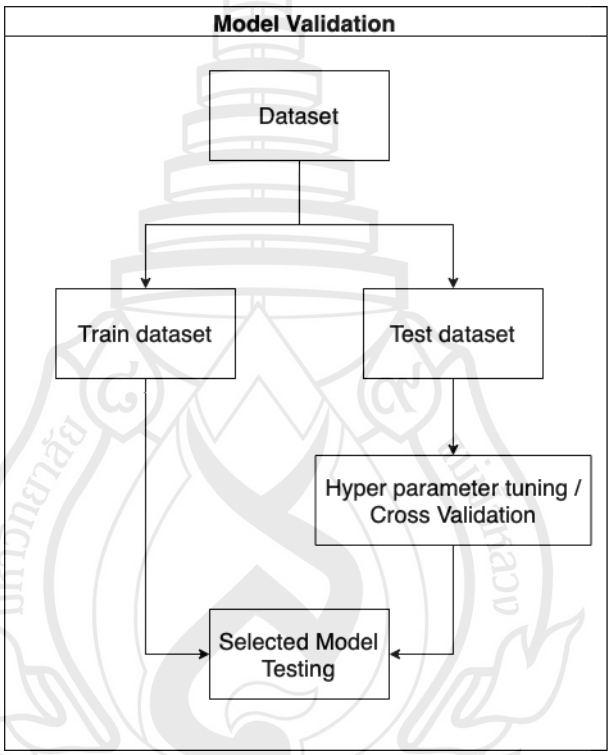


Figure 3.8 model validation flow

In this study used K-fold cross-validation with hyperparameter tuning. The dataset is divided into k folds, and the training and testing are repeated k times, with each fold serving as the testing set once. Hyperparameter tuning refers to the configuration settings of a model that are not learned from the data but set by the user, for instance, the number of trees in a random forest.

In this study used grid search methods for hyperparameter tuning. Grid search tries all possible combinations of hyperparameters within predefined ranges.

Cross-validation for hyperparameter tuning: during the hyperparameter tuning process, cross-validation is used to estimate the performance of each hyperparameter combination. The performance metrics obtained are compared after evaluating the model using cross-validation for each hyperparameter combination. The combination that results in the best performance according to the evaluation metric of interest is selected with the optimal set of hyperparameters. Once the optimal hyperparameters are obtained, the model is trained using these settings on the entire training set. The final model is then evaluated on separate testing to estimate its performance on unseen data.

3.5 Model Comparison

Model comparison is a process of comparing machine learning models using evaluation metrics that are interested in, such as accuracy, precision, recall, f1-score, and auc-roc score. Model comparison is a crucial step in machine learning to assess the performance of different algorithms and techniques applied to a specific task or dataset. In this section, the performance of various machine learning models is compared using a range of evaluation metrics, including accuracy, precision, recall, F1-score, and AUC-ROC score. The choice of evaluation metrics is guided by the specific characteristics of the problem at hand. Accuracy measures the overall correctness of predictions, while precision focuses on the proportion of true positive predictions among all positive predictions. Recall measures the ability of the model to correctly identify all relevant instances, while the F1 score provides a balance between precision and recall. AUC-ROC score evaluates the model's ability to distinguish between different classes. Evaluated several machine learning models, including Random Forest (RF), Gradient Boosting (GB), Extra Trees (ET), Support Vector Machine (SVM), Decision Tree Classifier (DCT), and Long Short-Term Memory (LSTM). Each model was trained and tested on the dataset with feature engineering alone and the dataset with feature engineering.

CHAPTER 4

METHODOLOGY

4.1 Research Overview

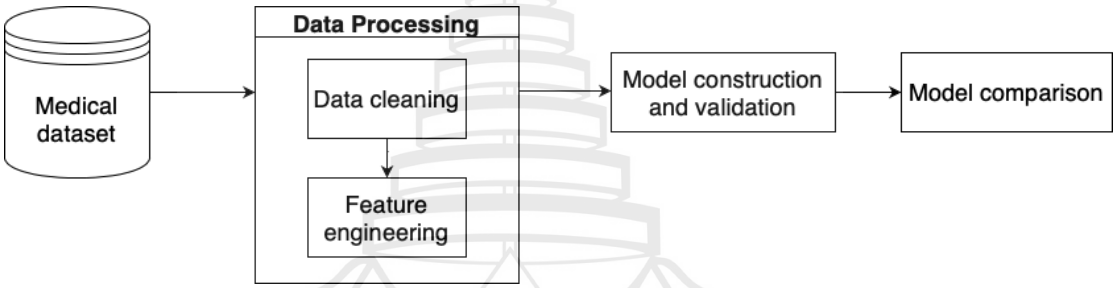


Figure 4.1 Diagram shows all the methodologies used in the process

This proposed study aims to develop a multiclass classification model for diabetes, hypertension, and the combination of diabetes with hypertension by using feature engineering tactics from previous work and handling the imbalanced dataset by combining it with CTGAN (Conditional Tabular GAN). This section will present all the processes. Figure 4.1 shows the entire process that will use in this study.

Starting with data collection, which is gathering data for this study, data preprocessing in this step will include data cleaning, which will handle the outlier within this dataset, and feature engineering techniques for manipulating the dataset to improve the performance of machine learning models. Model construction and validation will involve building the model incorporating CTGAN, validating the model, and finally, comparing models using score metrics to determine the best one.

4.2 Data Collection

In this study, more patient data were obtained from a local hospital, Phaya Menrai Hospital and Theong Hospital, Chiang Rai, Thailand; the dataset contained 58,463 samples. There were 14,461 samples of hypertension (24.74%), 42,800 samples of diabetes (73.2%), and 1,202 samples of both diseases (2.06%). There were 25 features. The sample of the dataset is provided in Table 4.1. This study received ethical approval from the Mae Fah Luang University Ethics Committee.

Table 4.1 All features of the dataset and its type

Data Type	Features
Ordinal	Smoking behavior
	Drinking behavior
	Education level
Nominal	Gender
	Family History
	Occupation
	Blood–Brain Barrier
Continuous	Age
	Height
	Body Mass Index
	Waist
	Blood Pressure Systolic
	Blood Pressure Diastolic
	Fasting Blood Sugar
	Total Cholesterol
	Triglyceride

Table 4.1 (continued)

Data Type	Features
Continuous	High-Density Lipoprotein
	Low-Density Lipoprotein
	Creatine
	Glomerular Filtration Rate
	Potassium
	Temperature
	Pulse
	Blood Urea Nitrogen
	Weight

4.3 Data Preprocessing

Preprocessing involves transforming raw data into a suitable format for machine learning algorithms. This study follows the previous one by starting with data cleansing and using the Interquartile Range (IQR), which is used for handling outliers in a dataset. It also treats features differently depending on their characteristics. In this study, they were grouped into three groups: ordinal, nominal, and continuous.

4.3.1 Ordinal Features

Feature improvement and feature construction for ordinal features In this study, use arbitrary imputation to handle missing data by inserting an arbitrary value to indicate the missingness of the data. Then, use the ordinal-encoding technique to transform categorical features into numerical representations. It assigns a unique numerical label to each variable category to preserve the ordinal relationship.

Table 4.2 Ordinal features and feature engineering methods

Data Type	Methods	Features
Ordinal	Arbitrary Value Imputation	Smoking behavior
	Ordinal Encoding	Drinking behavior
		Education level

4.3.2 Nominal Features

This study used a feature improvement technique called most frequent value imputation. Imputing the most repeated value can preserve the distribution by minimizing information loss since the imputed values are based on the observed value. Feature construction uses one-hot encoding, representing categorical variables as binary features that maintain categorical information.

Table 4.3 Nominal features and feature engineering methods

Data Type	Methods	Features
Nominal	Most-Frequent imputation	Gender
	One-hot encoding	Occupation
		Family History
		Blood–Brain Barrier

4.3.3 Feature Engineering for Continuous Features

This study used the Yeo-Johnson transformation to transform continuous features to achieve a more Gaussian-like distribution. Yeo-Johnson can handle zero and negative values. Then, this study used end-tail imputation to impute missing data with extreme values from the tails of the distribution.

The continuous feature represents numerical variables with a wide range of possible values. Table 3.4 presents a list of continuous features along with methods.

Table 4.4 Continuous features and feature engineering methods

Data Type	Methods	Features
Continuous	Normalization	Age
	Yeo-Johnson transformation	Height
	End-tail imputation	Body Mass Index
		Waist
		Blood Pressure Systolic
		Blood Pressure Diastolic
		Fasting Blood Sugar
		Total Cholesterol
		Triglyceride
		High-Density Lipoprotein
		Low-Density Lipoprotein
		Creatine
		Glomerular Filtration Rate
		Potassium
		Temperature
		Pulse
		Blood Urea Nitrogen
		Weight

4.4 Model Construction and Validation

This model construction and validation process were shown in figure 4.2

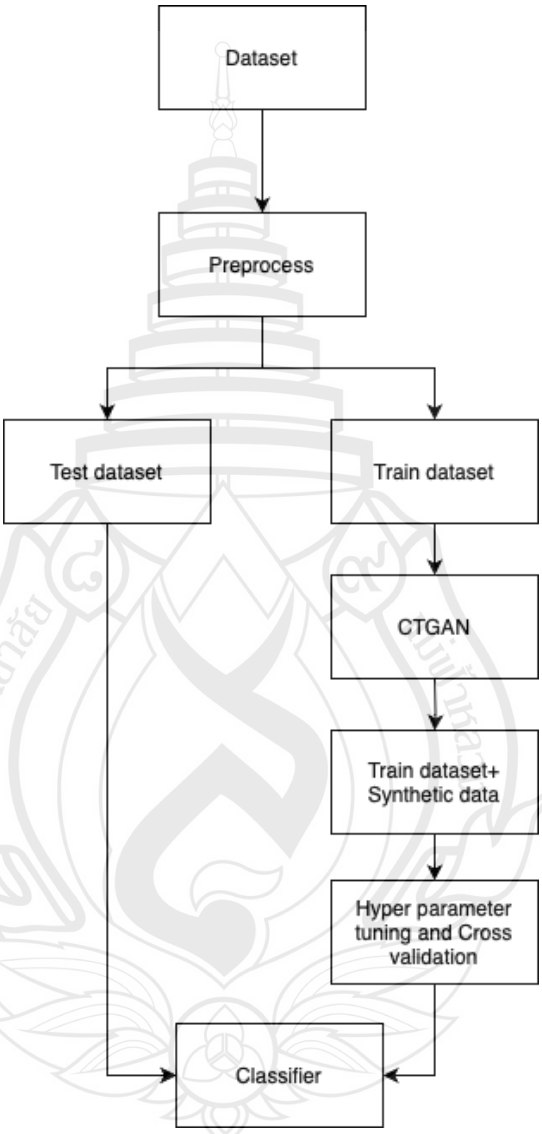


Figure 4.2 Model Construction and validation process overview

This study used the Yeo-Johnson transformation to transform continuous features to achieve a more Gaussian-like distribution. Yeo-Johnson can handle zero and negative values. Then, this study used end-tail imputation to impute missing data with extreme values from the tails of the distribution.

This study created a multiclass classification model and used three types of classifiers: traditional supervised learning, which are Decision Tree (DCT) and support vector machines (SVM); ensemble supervised learning, which are Random Forest (RF), Gradient

4.5 Model Comparison

Model comparison is a process of comparing machine learning models using evaluation metrics that are interested in, such as accuracy, precision, recall, f1-score, and auc-roc score. Model comparison is a crucial step in machine learning to assess the performance of different algorithms and techniques applied to a specific task or dataset. This section compares the performance of various machine learning models using a range of evaluation metrics, including accuracy, precision, recall, F1-score, and AUC-ROC score. The specific characteristics of the problem guide the choice of evaluation metrics. Accuracy measures the overall correctness of predictions, while precision focuses on the proportion of true positive predictions among all positive predictions. Recall measures the ability of the model to correctly identify all relevant instances, while the F1 score provides a balance between precision and recall. AUC-ROC score evaluates the model's ability to distinguish between different classes. evaluated several machine learning models, including Random Forest (RF), Gradient Boosting (GB), Extra Trees (ET), Support Vector Machine (SVM), Decision Tree Classifier (DCT), and Long Short-Term Memory (LSTM). Each model was trained and tested on the dataset with feature engineering alone and the dataset with feature engineering combined with CTGAN.

CHAPTER 5

RESULT AND DISCUSSION

The previous study used dataset 1, obtained from Phaya Mengrai Hospital, which has an imbalanced dataset issue. Tables 5.1 and 5.1 show the multiclass classification model without using the feature engineering score, and Tables 5.2 and 5.2 show the multiclass classification model's score using feature engineering.

Table 5.1 Dataset 1 baseline result

Model	Dataset 1 baseline				
	Accuracy	**Precision**	**Recall**	**F1-score**	**AUC-ROC**
RF	85.19%	90.868	85.13%	87.82%	90.12%
GB	84.99%	90.92%	85.36%	87.95%	89.36%
ET	80.74%	92.183	85.19%	88.15%	90.03%
SVM	85.02%	90.96%	84.08%	87.32%	88.17%
DCT	83.25%	89.97%	83.99%	86.85%	85.43%
LSTM	83.28%	90.41%	83.28%	86.60%	86.42%

Table 5.2 Dataset 1 with feature engineering

Model	Dataset 1 with feature engineering				
	Accuracy	**Precision**	**Recall**	**F1-score**	**AUC-ROC**
RF	88.07%	94.43%	88.51%	91.22%	93.02%

Table 5.2 (continued)

Model	Dataset 1 baseline				
	Accuracy	**Precision**	**Recall**	**F1-score**	**AUC-ROC**
GB	87.27%	94.31%	87.64%	90.84%	92.26%
ET	85.14%	92.96%	84.82%	88.33%	92.03%
SVM	88.39%	94.17%	87.67%	90.80%	93.32%
DCT	85.93%	92.31%	86.93%	89.38%	89.55%
LSTM	85.41%	92.62%	85.41%	88.76%	89.92%

According to Table 5.1 and Table 5.2, it can improve all classifiers in all metrics when combined with feature engineering. The promising result would be that SVM with feature engineering could beat other models based on accuracy, with the AUC-ROC metric score achieving a value of 88.39% and 93.32%, respectively. Ensemble learning methods Random Forest also showed a promising result. Decision performed the worst, but most importantly, all models performed better when feature engineering was applied.

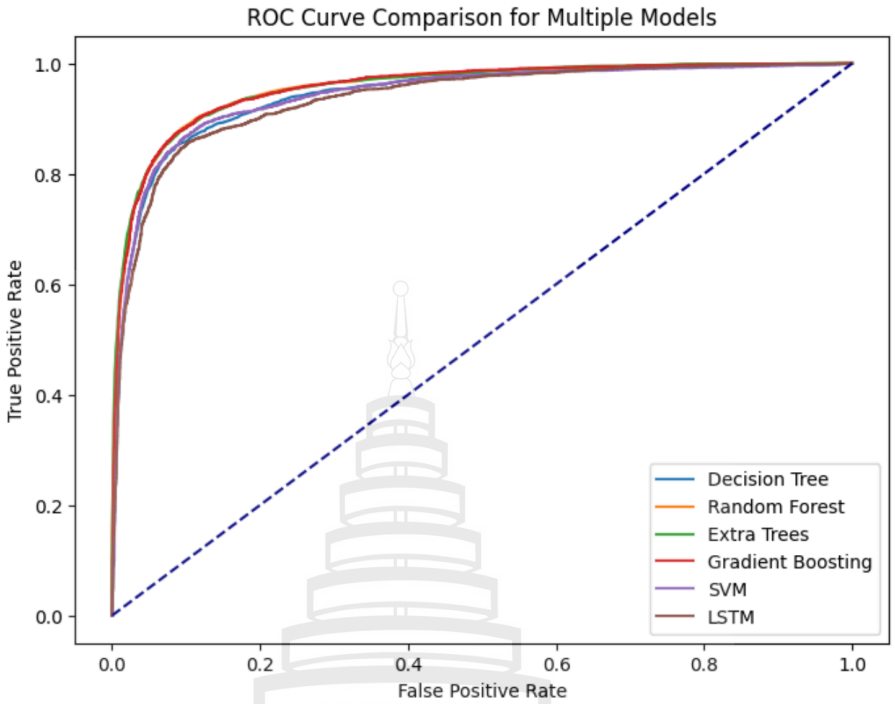


Figure 5.1 ROC curve comparison for dataset 1 without feature engineering

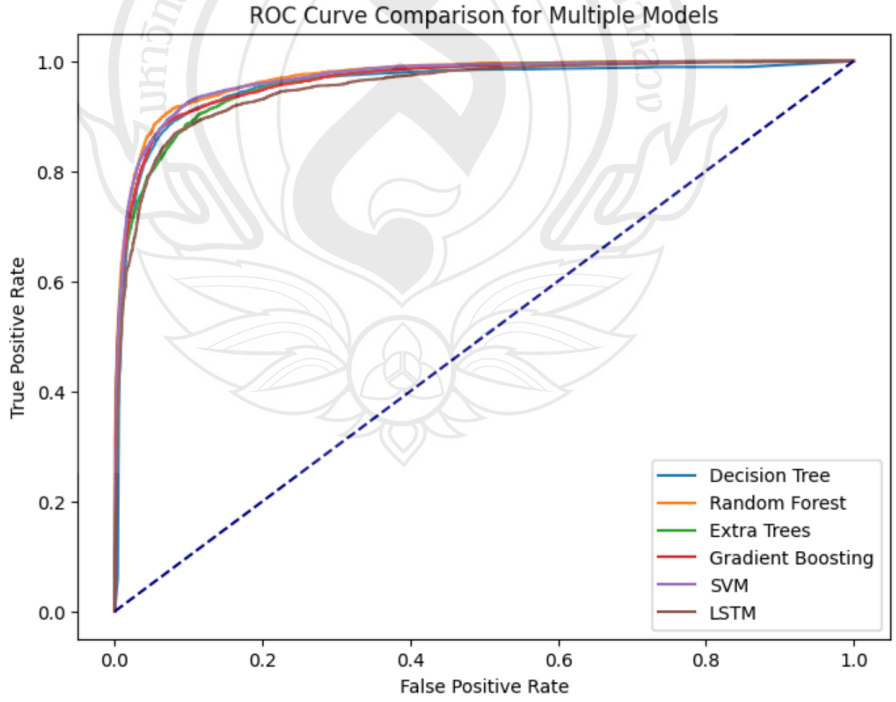


Figure 5.2 ROC curve comparison for dataset 1 with feature engineering

Figures 5.1 and 5.2 show the ROC of all classifiers. They show that all models perform pretty well for all groups, with DCT performing the worst. SVM and Ensemble learning outperform DCT. Based on this result, SVM and the ensemble learning methods are reasonably effective models for the multi-class prediction of hypertension with diabetes.

SVM outperformed other algorithms. SVM aims to maximize the margin between different classes, which can be beneficial when dealing with imbalanced classes. In a previous study, with hypertension being the majority class, SVM might be effectively finding a decision boundary that can separate hypertension from the other classes even though the dataset is imbalanced. Hyperparameter tuning and choice of the kernel also play a big role in enabling the model to handle imbalance to some extent by using the RBF kernel. Generalization: SVM is known for its ability to generalize well, which is crucial in cases of class imbalance. Even if the hypertension class is the majority, SVM still makes accurate predictions about the minority classes by creating an adequate margin. Lack of ensemble bias, on the other hand, ensemble learning methods, while often effective for class imbalance, can sometimes suffer from ensemble bias; if the imbalance dataset is too severe, it may struggle with the minority classes despite hyperparameter tuning. SVM, being a single model, might avoid this ensemble bias issue.

The next part will show the result after using dataset 2, from which more data were obtained from Phaya Mengrai Hospital and Theong Hospital. This proposed study results includes a feature correlation matrix, CTGAN metric results, results with feature engineering, and results with feature engineering and CTGAN.

Figure 5.3 below shows a feature correlation matrix, which shows us the relationship between features.

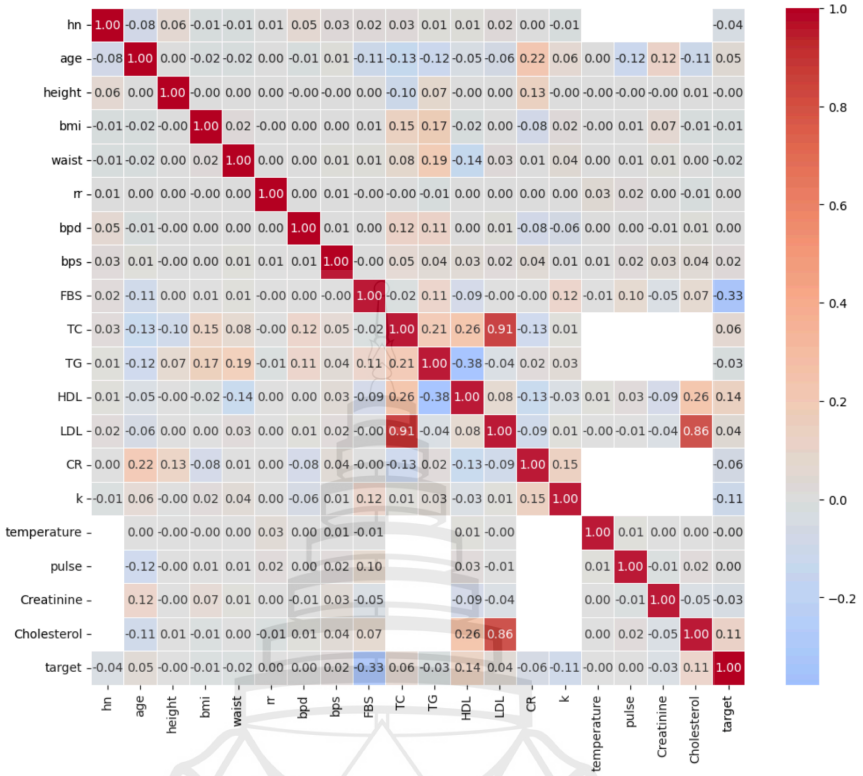


Figure 5.3 Correlation matrix

FBS shows a moderate negative correlation with the target variable, which suggests a notable relationship in which higher FBS might be associated with lower target values. TC, Age, CMI, and Waist Circumference show very weak correlations with the target, implying these variables might have minimal impact on predicting the target variable.

Previous work didn't handle imbalanced datasets, so in this study, the current proposed research incorporated CTGAN to address the imbalanced dataset issue and used the CTGAN metric to indicate how well CTGAN model is able to replicate the real dataset. The results are in Table 5.3 below.

Table 5.3 CTGAN metrics score

Metric	Score
Data validity score	100.0%
Data structure score	100.0%
Column shapes score	84.0%
Column Pair Trends Score	73.5%

According to Table 5.3, data validity is 100.0%. The model successfully generates structurally valid data that complies with the original data's basic rules and constraints. A data structure score of 100.0% means the synthetic data mirrors the structural format of the original data exactly, which is a positive sign. A score of 84.0% of Column shapes score indicates that synthetic data generally follows the same distributions as the original data. Lastly Column Pair Trends score of 73.5% indicates that the synthetic data capture most of the relationship between columns found in the original data.

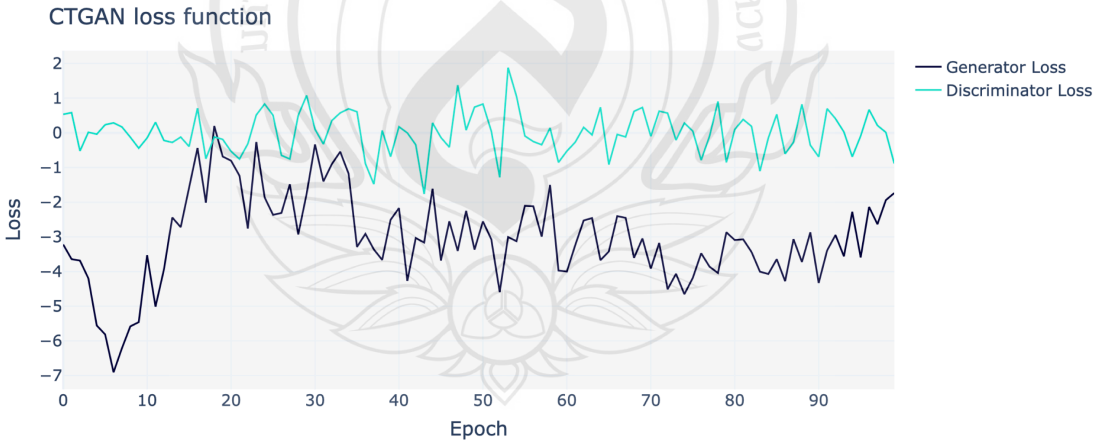


Figure 5.4 CTGAN loss function

According to Figure 5.4, the generator loss is lower than the discriminator loss. This suggests that the generator is doing a good job of creating synthetic data, which is

like real data. The discriminator is struggling to differentiate between real and generated data. The result from CTGAN performs well and replicates real datasets well.

According to a previous study, when combining a classification model with feature engineering, all machine learning algorithms can perform better. This proposed study is based on a previous study using feature engineering tactics and was able to obtain more data from Phaya Mengrai Hospital data and Theong Hospital, which is called dataset 2, and handle imbalanced dataset issues with CTGAN by generating minority class to have the same amount as majority class. Table 5.4 will show the result without CTGAN, and Table 5.5 will show the result with CTGAN.

Table 5.4 Dataset 2 with feature engineering

Model	Dataset 2 baseline with feature engineering				
	Accuracy	**Precision**	**Recall**	**F1-score**	**AUC-ROC**
RF	91.28%	93.68%	91.28%	92.46%	95.06%
GB	92.21%	93.88%	92.21%	93.04%	96.46%
ET	89.47%	91.34%	89.47%	90.19%	94.52%
SVM	90.73%	92.92%	90.73%	91.81%	93.98%
DCT	90.81%	92.84%	90.81%	91.80%	93.11%
LSTM	88.23%	89.81%	88.23%	88.95%	92.90%

Table 5.5 Dataset 2 with feature engineering and CTGAN

Model	Dataset 2 with feature engineering and CTGAN				
	Accuracy	**Precision**	**Recall**	**F1-score**	**AUC-ROC**
RF	91.65%	93.64%	91.65%	92.66%	95.08%
GB	92.08%	93.57%	92.08%	92.75%	95.85%
ET	91.17%	92.85%	91.18%	91.96%	94511%
SVM	87.13%	86.94%	87.13%	87.03%	87.63%
DCT	90.17%	92.78%	90.71%	91.64%	93.05%

Table 5.5 (continued)

Model	Dataset 1 baseline				
	Accuracy	**Precision**	**Recall**	**F1-score**	**AUC-ROC**
LSTM	78.15%	78.20%	78.15%	76.83%	83.51%

According to Table 5.4 and Table 5.5, when using CTGAN to handle an imbalanced dataset, there is no difference in terms of improvement. Only some classifiers are able to perform better when applied with CTGAN. The classifier with the greatest improvement when incorporated with CTGAN is Extra Trees.

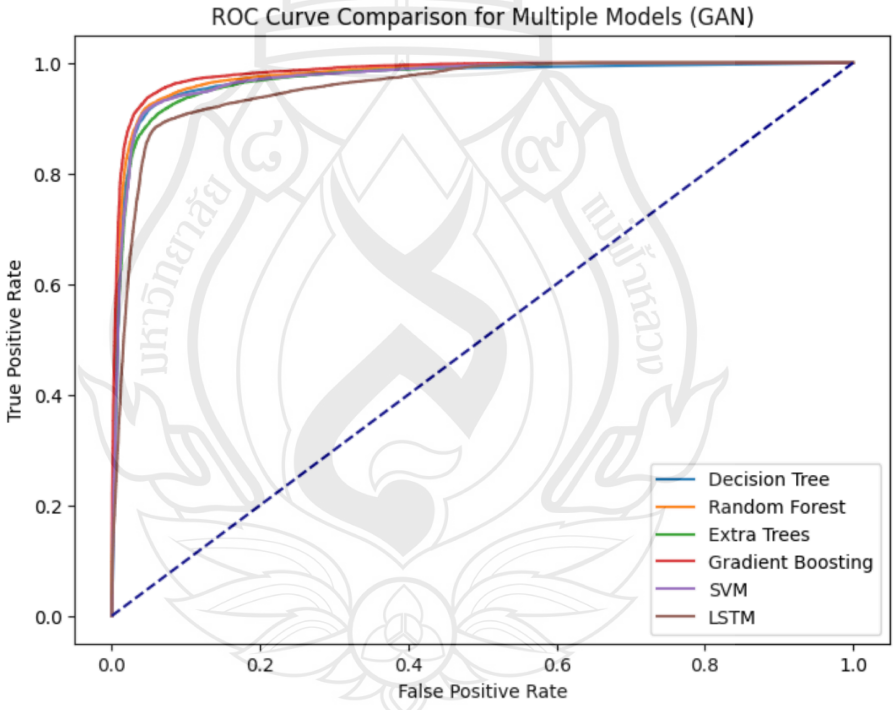


Figure 5.5 ROC curve comparison for dataset 2 with feature engineering

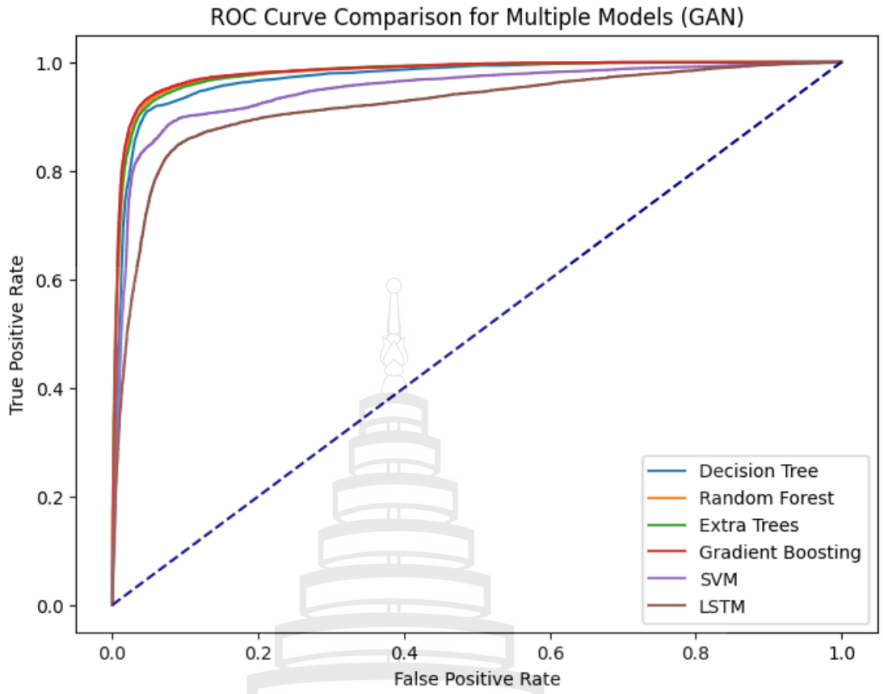


Figure 5.6 ROC curve comparison for dataset 2 with feature engineering and CTGAN

Figure 5.5 and Figure 5.6 shows the ROC of all classifiers. The figures show that all models perform well, LSTM seems to perform worst compared to other models. An interesting point is Extra Trees, in which some metrics can perform better than those without CTGAN, and LSTM performs worse than those without CTGAN.

Extra Trees are prone to overfit. The additional synthetic data from CTGAN helps to reduce overfitting by introducing more diversity into the dataset, which results in improved generalization. The introduction of synthetic data by CTGAN provides more varied training instances for each tree, leading to a more diverse ensemble and potentially better performance. CTGAN generates synthetic samples for minority classes. This can help Extra Trees better learn the decision boundaries for imbalanced classes, leading to improved performance.

Long-Short-Term Memory (LSTM) Performance declines with CTGAN. The LSTM model is designed for sequential data (time series) where the order of the input data is important. CTGAN generates synthetic samples independently of any sequential

dependencies present in the original data. The lack of sequentially in the synthetic data may not align well with LSTM, leading to a decrease in performance.

The feature engineering techniques may already provide sufficient information for the classifiers to learn effectively, reducing the additional benefit of CTGAN.

CHAPTER 6

CONCLUSION

Based on this study developed a multiclass classification model for hypertension, diabetes, and coexistence, and the models have shown promising results. Facing outliers, missing values and imbalanced issues commonly found in tabular datasets. Use feature engineering to overcome outlier and missing value issues. Classifiers with feature engineering can perform better when compared with models without feature engineering. For the imbalanced dataset, use CTGAN to generate a minority class to handle the imbalance issue. Even though CTGAN is used to handle the imbalance dataset issue, only one classifier can perform better: the Extra Trees classifier

There were two datasets. The previous work used the first dataset to see the difference between without feature engineering and with feature engineering, and the most promising result came from the SVM classifier with feature engineering, which achieved an 88.39% accuracy score and 93.36% AUC-ROC score. The second dataset used feature engineering, which showed promising results based on the previous studies and tried to see the difference between feature engineering and CTGAN. The most promising result came from Gradient Boosting without CTGAN, which achieved 92.21% accuracy and 96.46% AUC-ROC score. Most of the classifiers that were tried showed with or without CTGAN didn't have any big improvement; only Extra Trees were able to improve from 89.47% accuracy to 91.17% accuracy.

มหาวิทยาลัย
แม่ฟ้าหลวง

REFERENCES

REFERENCES

[1] World Health Organization(WHO). (2022, September). *Non communicable diseases.* https://www.who.int/news-room/fact-sheets/detail/noncommunicable-diseases

[2] Cheung, B. M. (2010). The hypertension–diabetes continuum. *Journal of Cardiovascular Pharmacology, 55*(4), 333–339.

[3] Naha, S., Gardner, M., Khangura, D., Kurukulasuriya, L., & Sowers, J. (2021). *Hypertension in diabetes*. https://www.ncbi.nlm.nih.gov/books/NBK279027

[4] Landsberg, L., & Molitch, M. (2004). Diabetes and hypertension: Pathogenesis, prevention and treatment. *Clinical and Experimental Hypertension, 26*(7-8), 621–628.

[5] Rachata, N., & Temdee, P. (2021). Mobile-based self-monitoring for preventing patients with type 2 diabetes mellitus and hypertension from cardiovascular complication. *Wireless Personal Communications, 117*(1), 151–175.

[6] Rajatanavin, N., Witthayapipopsakul, W., Vongmongkol, V., Saengruang, N., Wanwong, Y., Marshall, A. I., . . . Tangcharoensathien, V. (2021). Thailand effective coverage of diabetes and hypertension: Challenges and solutions. *medRxiv*. https://doi.org/10.1101/2021.03.22.21254093

[7] Hewett, M. L. (2010). Q: What is hypertension?. *Journal of the American Academy of PAs, 23*(7), 45–46.

[8] Strasser, T. (1992). The menace of high blood pressure. *World Health, January-February*, 12–13.

[9] World Health Organization(WHO). (2021, August). *Hypertension.* https://www.who.int/news-room/fact-sheets/detail/hypertension

[10] Viedma, C. (1991, May-June). *What is diabetes?.* World Health Organization. https://link.gale.com/apps/doc/A11083636/PPNU?u=thmfu&sid=bookmark-PPNU&xid=b8902bb3

[11] Deshpande, A. D., Harris-Hayes, M., & Schootman, M. (2008). Epidemiology of diabetes and diabetes-related complications. *Physical Therapy, 88*(11), 1254–1264.

[12] de Boer, I. H., & DCCT/EDIC Research Group. (2014). Kidney disease and related findings in the diabetes control and complications trial/Epidemiology of diabetes interventions and complications study. *Diabetes Care, 37*(1), 24–30.

[13] Kurkela, O., Nevalainen, J., Arffman, M., Lahtela, J., & Forma, L. (2022). Foot-related diabetes complications: Care pathways, patient profiles and costs. *BMC Health Services Research, 22*(1), 1–11.

[14] World Health Organization(WHO). (2021, November). *Diabetes*. https://www.who.int/news-room/fact-sheets/detail/diabetes

[15] Mayo Clinic. (2020, October). Diabetes - Diagnosis and treatment. https://www.mayoclinic.org/diseases-conditions/diabetes/diagnosis-treatment/drc-20371451

[16] Chen, M., Hao, Y., Hwang, K., Wang, L., & Wang, L. (2017). Disease prediction by machine learning over big data from healthcare communities. *IEEE Access, 5*, 8869–8879.

[17] Barakat, N., Bradley, A. P., & Barakat, M. N. H. (2010). Intelligible support vector machines for diagnosis of diabetes mellitus. *IEEE Transactions on Information Technology in Biomedicine, 14*(4), 1114–1120.

[18] Mohan, S., Thirumalai, C., & Srivastava, G. (2019). Effective heart disease prediction using hybrid machine learning techniques. *IEEE Access, 7*, 81542–81554.

[19] Tabik, S., Gómez-Rias, A., Martín-Rodríguez, J.L., Sevillano-García, I., Rey-Area, M., Charte, D., . . . Herrera, F. (2020). COVIDGR dataset and COVID-SDNet methodology for predicting COVID-19 based on chest X-ray images. *IEEE Journal of Biomedical and Health Informatics, 24*(12), 3595–3605.

[20] Shin, H. C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., . . . Summers, R.M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics, and transfer learning. *IEEE Transactions on Medical Imaging,35*(5), 1285–1298.

[21] Vinodhini, V., Vishalakshi, A., Chandrika, G. N., Sankar, S., & Ramasubbareddy, S. (2022). Predicting vasovagal syncope for paraplegia patients using average weighted ensemble technique. *Journal of Mobile Multimedia, 18*(3), 135–162.

[22] Sankar, S., Potti, A., Chandrika, G. N., & Ramasubbareddy, S. (2022). Thyroid disease prediction using XGBoost algorithms. *Journal of Mobile Multimedia, 18*(3), 1–18.

[23] Prasad, J. V. D., Pratap, A. R., & Sallagundla, B. (2021). Machine learning-based clinical diagnosis of liver patients with instance replacement. *Journal of Mobile Multimedia, 18*(2), 293–306.

[24] Das, S., Amoedo, B., De la Torre, F., & Hodgins, J. (2012). Detecting Parkinson's symptoms in uncontrolled home environments: A multiple instance learning approach. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 3688–3691). IEEE.

[25] Alian, S., Li, J., & Pandey, V. (2018). A personalized recommendation system to support diabetes self-management for American Indians. *IEEE Access, 6*, 73041–73051.

[26] Khan, A., Doucette, J. A., Cohen, R., & Lizotte, D. J. (2012). Integrating machine learning into a medical decision support system to address the problem of missing patient data. In *2012 11th International Conference on Machine Learning and Applications* (Vol. 1, pp. 454–457). IEEE.

[27] Pitoglou, S., Koumpouros, Y., & Anastasiou, A. (2018). Using electronic health records and machine learning to make medical-related predictions from non-medical data. In *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)* (pp. 56–60). IEEE.

[28] MedlinePlus. (2022, July 30). *High blood pressure.* https://medlineplus.gov/highbloodpressure.html

[29] Lama, L., Wilhemsson, O., Norlander, E., Gustafsson, L., Lager, A., Tynelius, P., . . . Ostenson, C.G. (2021). Machine learning for prediction of diabetes risk in middle-aged Swedish people. *Heliyon, 7*(7), e07419.

[30] Mirzajani, S. S. (2018). Prediction and diagnosis of diabetes by using data mining techniques. *Avicenna Journal of Medical Biochemistry, 6*(1), 3–7.

[31] Sonar, P., & Jayamalini, K. (2019). Diabetes prediction using different machine learning approaches. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 367–371). IEEE.

[32] Rahman, M., Islam, D., Mukti, R. J., & Saha, I. (2020). A deep learning approach based on convolutional LSTM for detecting diabetes. *Computational Biology and Chemistry*, 88, https://doi.org/10.1016/j.compbiolchem.2020.107329

[33] Butt, U. M., Letchmunan, S., Ali, M., Hassan, F. H., Baqir, A., & Sherazi, H. H. R. (2021). Machine learning-based diabetes classification and prediction for healthcare applications. *Journal of Healthcare Engineering*, *2021*(1), 9930985

[34] S., G., R., V., & S., K. P. (2018). Diabetes detection using deep learning algorithms. *ICT Express*, *4*(4), 243–246.

[35] Nasir, N., Oswald, P., Barneih, F., Alshaltone, O., Alshabi, M., Bonny, T., . . . Al Shammaa, A. (2021). Hypertension classification using machine learning part II. In *2021 14th International Conference on Developments in eSystems Engineering (DeSE)* (pp. 459–463). IEEE.

[36] AlKaabi, L. A., Ahmed, L. S., Al Attiyah, M. F., & Abdel-Rahman, M. E. (2020). Predicting hypertension using machine learning: Findings from Qatar Biobank Study. *PLOS ONE, 15*(10), e0240370.

[37] Jain, K., Jha, J., & Jha, Y. (2021). Comparative analysis of machine learning algorithms for blood pressure prediction. In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 636–642). IEEE.

[38] Yen, C.-T., Chang, S.-N., & Liao, C.-H. (2021). Deep learning algorithm evaluation of hypertension classification in less photoplethysmography signals conditions. *Measurement and Control, 54*(3-4), 439–445. SAGE Publications.

[39] Choi, Y. Y., Jeong, H., Lee, J. H., Sung, K. C., Shin, J.-H., Kim, H. C., . . . Kang, D. R. (2021). Cardiovascular disease prediction model in patients with hypertension using deep learning: Analysis of the National Health Insurance Service database from Republic of Korea. *CardioMetabolic Syndrome Journal, 1*(3), 145. Korean Society of CardioMetabolic Syndrome.

[40] Fitriyani, N. L., Syafrudin, M., Alfian, G., & Rhee, J. (2019). Development of disease prediction model based on ensemble learning approach for diabetes and hypertension. *IEEE Access*, 7, 144777–144789.

[41] Tabosa de Oliveira, T., da Silva Neto, S. R., Teixeira, I. V., Aguiar de Oliveira, S. B., de Almeida Rodrigues, M. G., Sampaio, V. S., . . . Endo, P. T. (2022). A comparative study of machine learning techniques for multi-class classification of arboviral diseases. *Frontiers in Tropical Diseases*, 2. https://doi.org/10.3389/fitd.2021.769968

[42] Khan, M. A., Majid, A., Hussain, N., Alhaisoni, M., Zhang, Y.-D., Kadry, S., . . . Nam, Y. (2021). Multiclass stomach diseases classification using deep learning features optimization. Computers*, Materials & Continua, 67*(3), 3381–3399. https://doi.org/10.32604/cmc.2021.014983

[43] Habibi, O., Chemmakha, M., & Lazaar, M. (2023). Imbalanced tabular data modelization using CTGAN and machine learning to improve IoT Botnet attacks detection. *Engineering Applications of Artificial Intelligence, 118*, 105669. https://doi.org/10.1016/j.engappai.2022.105669

[44] Jia, J., Wu, P., & Dawood, H. (2023). An improved CTGAN for data processing method of imbalanced disk failure. *arXiv* preprint arXiv:2310.06481. https://doi.org/10.48550/ARXIV.2310.06481

[45] Wang, J., Yan, X., Liu, L., Li, L., & Yu, Y. (2022). CTTGAN: Traffic data synthesizing scheme based on conditional GAN. *Sensors, 22*(14), 5243. https://doi.org/10.3390/s22145243

[46] Majeed, A., & Hwang, S. O. (2023). CTGAN-MOS: Conditional generative adversarial network based minority-class-augmented oversampling scheme for imbalanced problems. *IEEE Access*, *11*, 85878–85899. https://doi.org/10.1109/access.2023.3303509

[47] Nair, P., & Kashyap, I. (2019). Hybrid pre-processing technique for handling imbalanced data and detecting outliers for KNN classifier. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)* (pp. 460–464). Faridabad, India.

[48] Ozedemir, S. (2022). *Feature Engineering Bookcamp*. Manning Publications.

[49] Raymaekers, J., & Rousseeuw, P. J. (2021). Transforming variables to central normality. *Machine Learning*, 1–23.

[50] Serrano, L. G. (2023). *Grokking Machine Learning*. Manning.

[51] Kunapuli, G. (2023). *Ensemble Methods for Machine Learning*. Manning Publications.

[52] Raff, E. (2022). *Inside Deep Learning*. Manning.

[53] Mark, L. (2024). *Learn Generative AI with PyTorch*. Manning.

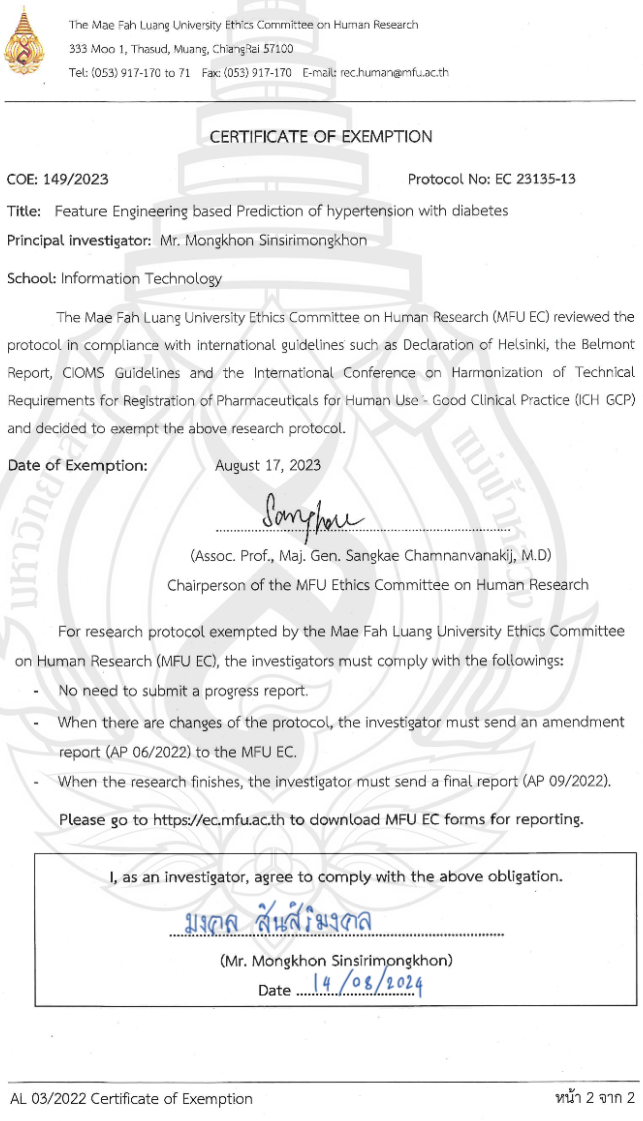
[54] Xu, L., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2019). Modeling tabular data using conditional GAN. *Advances in neural information processing system,32*. https://doi.org/10.48550/ARXIV.1907.00503

มหาวิทยาลัย
แม่ฟ้าหลวง

APPENDIX

APPENDIX

ETHICAL APPROVAL CERTIFICATE



The Mae Fah Luang University Ethics Committee on Human Research
333 Moo 1, Thasud, Muang, ChiangRai 57100
Tel: (053) 917-170 to 71 Fax: (053) 917-170 E-mail: rec.human@mfu.ac.th

CERTIFICATE OF EXEMPTION

COE: 149/2023 Protocol No: EC 23135-13

Title: Feature Engineering based Prediction of hypertension with diabetes

Principal investigator: Mr. Mongkhon Sinsirimongkhon

School: Information Technology

The Mae Fah Luang University Ethics Committee on Human Research (MFU EC) reviewed the protocol in compliance with international guidelines such as Declaration of Helsinki, the Belmont Report, CIOMS Guidelines and the International Conference on Harmonization of Technical Requirements for Registration of Pharmaceuticals for Human Use - Good Clinical Practice (ICH GCP) and decided to exempt the above research protocol.

Date of Exemption: August 17, 2023

(Assoc. Prof., Maj. Gen. Sangkae Chamnanvanakij, M.D)
Chairperson of the MFU Ethics Committee on Human Research

For research protocol exempted by the Mae Fah Luang University Ethics Committee on Human Research (MFU EC), the investigators must comply with the followings:

- No need to submit a progress report.
- When there are changes of the protocol, the investigator must send an amendment report (AP 06/2022) to the MFU EC.
- When the research finishes, the investigator must send a final report (AP 09/2022).

Please go to https://ec.mfu.ac.th to download MFU EC forms for reporting.

I, as an investigator, agree to comply with the above obligation.

มงคล สินสิริมงคล

(Mr. Mongkhon Sinsirimongkhon)
Date 14/08/2024

AL 03/2022 Certificate of Exemption หน้า 2 จาก 2

Figure 1 Certificate of Exemption