Evaluation of Data Preprocessing Strategy on the Robustness of the LightGBM Model in Hypertension Risk Prediction

Moch. Andri Setiawan[a], Muhammad Najibulloh Muzaki[b], Aidina Ristyawan[c]

[a,b,c]*Department of Information Systems, Universitas Nusantara PGRI Kediri, Indonesia*

ARTICLE INFO

Article history:
Received May 5, 2026
Received in revised form
May 29, 2026
Accepted June 3, 2026
Available online June 30, 2026

Keywords:
Data Preprocessing
Data Quality
Hypertension Prediction
LightGBM
Robustness Models

ABSTRACT

Hypertension is a chronic disease known as the "silent killer" because it often does not show early symptoms; therefore, early detection of the risk of hypertension is very important. Various studies have shown that the development of machine learning models generally focuses more on improving the accuracy of predictions without considering the quality of the data used. This study aimed to evaluate the influence of data quality, especially the existence of placeholder values, on the robustness of the LightGBM model through the application of preprocessing strategies. The methods used included hyperparameter optimization using randomized search cross-validation (CV), class weight adjustment, and the application of machine learning pipelines. The results showed that the model was able to maintain high performance with an accuracy of 99.69% on the original data, and the same accuracy of 99.46% on the cleaned and imputation data. In addition, the small performance difference between the schemes and an AUC value above 0.999 across all tests showed that the model had a stable classification performance with no indication of significant overfitting. Despite a slight drop in performance after placeholder cleanup and data imputation, the model still showed good stability and generalization. Thus, the proposed LightGBM model has a good level of robustness for predicting hypertension risk.

1 Introduction

Hypertension is a medical condition in which blood pressure against the artery walls is persistently high [1]. Hypertension was defined as a systolic blood pressure ≥ 140 mmHg or diastolic blood pressure ≥ 90 mmHg measured on two different occasions. However, some recent guidelines have set a lower threshold of ≥ 130/80 mmHg for early prevention [2]. According to the World Health Organization (WHO), approximately 1.28 billion adults worldwide suffer from hypertension, and almost 46% of them are unaware of the condition. Hypertension is known as the "silent killer" because it often shows no initial symptoms, but it can lead to serious complications, such as heart disease, stroke, and kidney disorders [3]. Globally, approximately 10.8 million deaths each year are related to hypertension [2].

In Indonesia, the prevalence of hypertension tends to increase with age, reaching 10.7% in the 18–24 years age group and 17.4% in the 25–34 years age group. Low public awareness of the need for routine health check-ups and delayed diagnosis are the main challenges in controlling hypertension [4]. Therefore, an approach that can support the early detection of hypertension quickly and accurately is needed.

With the development of technology, machine learning has been widely used in the health sector to aid in the disease prediction process [5]. Preliminary studies conducted by the authors showed that the LightGBM model with hyperparameter optimization could achieve an accuracy of 99.69%. In addition, several other studies using the "Arterial Hypertension Mexico" dataset have also reported high performance in cases of hypertension prediction. Research [6] showed that the CatBoost algorithm as the best model obtained an accuracy of 99.2%, while research [7] reported the AdaBoost algorithm as the best model with an accuracy of 97.60%.

However, most previous studies have focused on improving model performance through algorithm selection and hyperparameter optimization, while the data quality aspect is still relatively underpaid. In practice, health datasets often contain invalid values and anomalies that can affect the model's learning process [8]. In the "Arterial Hypertension Mexico" dataset, placeholder values were found, which are the values used as data markers that are not available or invalid and do not represent the actual medical condition. These values have the potential to form a pseudo-pattern so that the model can generate high performance numerically, but they do not fully reflect the predictive ability of real data conditions [9].

Based on previous research reviews, there is still a research gap related to the lack of evaluation of the influence of data quality and the handling of placeholder values on the performance of models for predicting hypertension. Most previous studies used the same dataset with a primary focus on improving the prediction performance. However, the aspect of data quality, especially the existence of placeholder values that have the potential to affect the model learning process, has not been systematically evaluated. Consequently, it is uncertain whether the reported high performance can still be maintained after the invalid value is handled through the preprocessing strategy.

Therefore, this study focuses on evaluating the stability and consistency of model performance against variations in data quality, particularly to analyze the extent to which the handling of placeholder values affects prediction results. This study used LightGBM as the main machine learning model in the modeling process. The focus of this study was not on comparing various algorithms to obtain the highest accuracy but on analyzing the influence of data quality and preprocessing strategies on model performance and robustness.

The main contributions of this study are as follows: (1) identification and evaluation of placeholder values in the "Arterial Hypertension Mexico" dataset that have the potential to affect the validity of the prediction model, (2) analysis of the influence of preprocessing strategies on the performance and robustness of the LightGBM model, and (3) application of machine learning pipelines to ensure that the preprocessing process is carried out consistently and minimize potential data leakage [10]. In contrast to previous studies that focused on achieving the highest accuracy using the same dataset, this study emphasizes the evaluation of data quality and the resilience of the model to changes in the data conditions. Thus, this study provides a more comprehensive understanding of the relationship between data quality, preprocessing strategies, and the robustness of the LightGBM model in predicting hypertension risk.

2 Literature Study

2.1 Hypertension

Hypertension is a medical condition characterized by persistently increased blood pressure and is a major risk factor for cardiovascular disease. This condition often shows no symptoms in the early stages, so it is known as the "silent killer." Early detection of hypertension is essential to prevent serious complications, such as stroke, heart disease, and kidney failure. Therefore, a data-driven approach is needed to help the hypertension risk identification process more effectively [2].

2.2 Data Mining and Machine Learning

Data Mining and Machine Learning are two interrelated concepts in the process of data analysis that produce valuable knowledge. Data Mining focuses on the process of extracting important patterns or information from large amounts of data as part of knowledge discovery in databases (KDD) [11]. Machine learning is a data mining approach that allows the system to learn

Corresponding Author: mochandrisetiawan2@gmail.com

Doi: 10.30996/elsains.v8i1.133544

from data to make predictions and decisions automatically. Various approaches, such as supervised and unsupervised learning, have been widely used in various fields, including health, because they can improve the efficiency and accuracy of data analysis [12].

2.3 Classification

Classification is a technique in machine learning that is used to group data into certain classes based on the patterns learned from the training data. The main goal of classification is to build a model capable of accurately predicting the labels of the new data. In general, the classification process is performed by selecting the class with the highest probability based on the features of the data [13].

2.4 Data Quality and Data Preprocessing

Data quality is an important factor in data mining and machine learning because it affects the performance, validity, and generalization of the model. Low-quality data, such as missing values, noise, inconsistencies, outliers, and invalid values, can decrease model accuracy and stability. Therefore, a data preprocessing stage is required to improve the quality of the data before it is used in the modeling process [14]. Data preprocessing is an important stage in data mining that aims to prepare and transform data to be suitable for analysis and modeling [15]. This stage not only improves the data quality but also reduces data problems and helps improve the model performance. In general, the preprocessing process includes data cleaning, integration, and transformation.

1) Data Cleaning: It process handles problems with data, such as missing values, invalid data, noise, outliers, and inconsistencies. This stage was performed to ensure that the data were clean, consistent, and reliable before being used in the modeling process.
2) Data Integration: It process of combining data from various sources into a unified and consistent dataset. The goal was to facilitate the analysis and ensure that there were no conflicts between the data.
3) Data Transformation: It process of converting data into a form that is suitable for modeling needs. This process can be in the form of normalization, standardization, and conversion of categorical data into numerical data so that they can be processed by machine learning algorithms.

2.5 Placeholder values

Placeholder values are used as markers for unavailable, unmeasurable, or invalid data. In health datasets, these values are often represented as specific numbers that do not reflect the actual condition of a variable. If not handled appropriately, placeholders can be considered valid data by machine learning algorithms, potentially forming spurious patterns and producing high performance numerically, but not reflecting predictive capabilities under real conditions [9].

2.6 Machine Learning Pipeline for Data Leakage Prevention

Pipelines in machine learning are structured workflows that systematically integrate all stages of data mining, from preprocessing and feature transformation to model training and evaluation. The use of pipelines aims to increase reproducibility, reduce implementation errors, and prevent data leakage by ensuring that the entire learning process uses only training data. Data leakage occurs when information from the test data or information directly related to the target variable is accidentally used during the model training process. This condition can cause the evaluation performance to be excessively high and not reflect the model's actual generalization capabilities. Therefore, all preprocessing stages, such as imputation, encoding, and normalization, must be carried out based on the training data and then applied to the test data using the parameters that have been studied previously [16].

The pipeline has two main processes: fit and transform. The fit process is a learning stage from the training data to obtain the necessary parameters, such as statistical values or model structures, which will be used in the next stage. Meanwhile, the transform process is the application of parameters that have been studied to consistently change the training and test data without relearning [10]. With this approach, each stage of preprocessing is carried out in isolation on the training data to minimize the risk of data leakage and result in a more valid model evaluation.

2.7 LightGBM Algorithm

The Light Gradient Boosting Machine (LightGBM) is an ensemble algorithm based on the Gradient Boosting Decision Tree (GBDT) designed to improve the efficiency and accuracy of large and high-dimensional data. LightGBM builds the model additively through the summation of the decision tree, which is expressed as

$$\hat{Y}_i = \sum_{m=1}^{M} f_m(x_i) \quad (1)$$

With $\hat{Y}_i$ as the predicted value for the i-th data point, x_i as the feature vector for that data, $f_m(x_i)$ as the representation of the m-th decision tree, and M as the total number of trees. In the training process, LightGBM applies a leaf-wise strategy, meaning that it grows the tree by expanding the leaf nodes and selecting the node that provides the most optimal improvement in performance.

This approach allows for a faster loss reduction than the level-wise method. In addition, LightGBM is equipped with a model complexity control mechanism to minimize the risk of overfitting [17].

2.8 K-Nearest Neighbors (KNN) Imputer

The KNN Imputer is a data imputation method used to handle missing values by utilizing the proximity or similarity between data. This method works by looking for the nearest neighbor K from an observation that has a missing value, and then the value is estimated based on information from the nearest neighbor found. The degree of similarity is calculated using Euclidean distances:

$$d\left(x_i, x_j\right) = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right)^2} \quad (2)$$

Where $d\left(x_i, x_j\right)$ denotes the distance between the i-th and j-th observations, x_{ik} and x_{jk} represent the value of the k-th feature of each observation, and n is the total number of features. After computing the distances between observations, the K nearest neighbors are selected. The missing value is then estimated as the mean of the K neighbors, defined as:

$$\hat{x}_i = \frac{1}{K} \sum_{j \in N(i)} x_j \quad (3)$$

Where $N(i)$ denotes the set of K nearest neighbors of the i-th observation, and x_j represents the available feature value of the j-th neighbor. For categorical data, the mode of the neighbors is used as the imputation value [5].

2.9 Model Evaluation

The model was evaluated to assess the performance of the model algorithm. One of the methods used is the confusion matrix, which presents information about the number of correct and false predictions based on a comparison between the actual class and the model prediction results [8], as shown in Table 1.

Table 1. Confusion Matrix

Classes	Prediction *Yes*	Prediction *No*	Total
Current *Yes*	*True Positive* (HCMC)	*False Negative* (FN)	*Positive* (P)
Current *No*	*False Positive* (FP)	*True Negative* (TN)	*Negative* (N)
Total	P’	N’	P + N

Table 1. It displays four main components in the confusion matrix: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The four components illustrate the comparison between the model's predicted results and the actual label. These values are the basis for calculating various evaluation metrics, such as accuracy, precision, recall, F1-score, and Area Under Curve (AUC) [6]. The following formula was used for the calculation:

Corresponding Author: mochandrisetiawan2@gmail.com
Doi: 10.30996/elsains.v8i1.133544

$$1) \quad Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$2) \quad \text{Precision} = \frac{TP}{TP+FP} \quad (5)$$

$$3) \quad Recall = \frac{TP}{TP+FN} \quad (6)$$

$$4) \quad F1\text{-}score = \frac{2 \times recall \times precision}{recall + precision} \quad (7)$$

$$5) \quad AUC = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \quad (8)$$

3 Methodology

This study used the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology as a framework for the development of hypertension prediction models. This methodology was chosen because it is systematic, flexible, and commonly used in data mining projects [18]. The complete flow of CRISP-DM is shown in Figure 1.

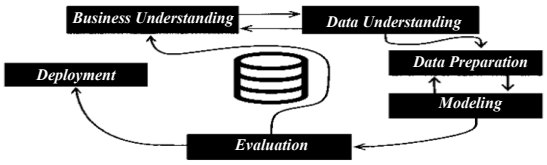

Figure 1. *CRISP-DM*

3.1 Business Understanding

The aim was to understand the problems and objectives of the research [18]. The main problem is that the limitations in identifying the risk of hypertension quickly and accurately remain a challenge. The purpose of this study was to develop an accurate and robust model for hypertension prediction.

3.2 Data Understanding

The aim was data collection and understanding [18]. This stage begins with the collection of data sourced from Kaggle, namely the "Mexican Arterial Hypertension" dataset [19]. The dataset consisted of 4,363 data points with 36 attributes. More information about the dataset is provided in Table 2.

Table 2. Dataset Information

Title	Remarks
Type Data	float64 (24), int64 (11), object (1)
Missing Value	None
Duplicate Data	None
Data Inconsistencies	None
Class Distribution	Risk: 2,816 (64.5%), No Risk: 1,547 (35.5%)
Target	*Hypertension* (1 = risk, 0 = No Risk)

This was followed by Exploratory Data Analysis (EDA) to structure the data, detect patterns, and identify anomalies. EDA is an important stage in the data mining process to understand the initial characteristics of the dataset before the modeling stage is carried out, including the identification of data quality and potential issues that could affect model performance [20].

The results of the initial examination showed that the dataset did not contain missing values, data duplication, or structural inconsistencies; therefore, there was no need for a data cleaning process in these aspects. However, placeholder values were found in 11 main features, indicating the use of substitution values in the data collection process. These placeholder values do not represent the results of actual clinical measurements but are used as markers for data that are not available or cannot be measured.

	Variabel	Nilai	Total Modus
0	peso	222.22	961 kali (22.03%)
1	estatura	222.2	24 kali (0.55%)
2	medida_cintura	222.2	1320 kali (30.25%)
3	segundamedicion_peso	222.22	6 kali (0.14%)
4	segundamedicion_estatura	222.2	8 kali (0.18%)
5	distancia_rodilla_talon	222.2	14 kali (0.32%)
6	circunferencia_pantorrilla	222.2	14 kali (0.32%)
7	segundamedicion_cintura	222.2	3493 kali (80.06%)
8	sueno_horas	99	3 kali (0.07%)
9	tension_arterial	999	29 kali (0.66%)
10	masa_corporal	1	943 kali (21.61%)

Figure 2. *Placeholder*

Based on Figure 2, the results of checking the placeholder values show that the segundamedicion_cintura variable has the highest proportion of 80.06%, followed by medida_cintura (30.25%), peso (22.03%), and masa_corporal (21.61%). Other variables also contain placeholder values with smaller proportions. These findings show that the existence of placeholder values is not minor but quite significant in some of the main anthropometric variables.

From a clinical and biomedical perspective, these values do not fully represent realistic human physiological conditions. For example, values of 222.2 and 222.22 for the anthropometric variables are outside the normal human biological range and consistently appear for some variables, indicating a pattern of substitution values. Similarly, a blood pressure value of 999 does not meet clinical standards based on the Joint National Committee (JNC) guidelines, where extreme values of blood pressure rarely exceed 250 mmHg [21]. In addition, the value of 99 for sleep duration does not correspond to the human physiological range, which is limited by circadian rhythms of approximately 24 h [22].

Furthermore, a masa_corporal value (BMI) of 1 kg/m² does not represent a valid human biological condition. Based on the World Health Organization (WHO) classification, a BMI value below 16 kg/m² is categorized as severe malnutrition, whereas a value of 1 kg/m² is physiologically incompatible with human life [23]. Therefore, these values were identified as system-generated placeholder values and not the results of actual clinical observations.

If these values are not identified appropriately, there is a significant risk in the modeling process. Tree-based algorithms such as LightGBM treat these values as valid numerical data, so that they can form spurious decision boundaries, which are classification decision boundaries formed by the model studying pseudo-patterns, data artifacts, or relationships that have no causal basis with the target variable. This can lead to two main consequences: (1) the formation of spurious patterns that affect the interpretation of feature importance, and (2) increased inflated performance due to easy separation of data at extreme values that are not clinically meaningful [9]. Therefore, the initial identification and analysis of data quality is a crucial stage before the modeling process is carried out.

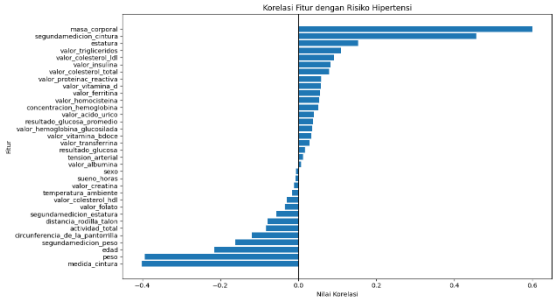

Figure 3. Correlation of Features with Target Class

Furthermore, correlation analysis was performed between hypertension target features and variables to evaluate the statistical relationship between variables while identifying potential data leakage. Based on Figure 3, no features were found that had a near-perfect correlation (±1) with the target. The highest correlation was obtained in masa_corporal features with a value of approximately 0.6, which is included in the category of moderate to strong correlation and is still clinically relevant as a risk factor for hypertension. These results show that there are no strong indications of features that directly represent the target variable.

In addition, masa_corporal (BMI), segundamedicion_cintura (measurement of second waist circumference), stature (height), and valor_colesterol_LDL showed relatively stronger positive

correlations with hypertension risk. In contrast, the medida_cintura (waist circumference), peso (weight), edad (age), and circunferencia_de_pantorrilla (calf circumference) features showed a more dominant negative correlation, although most of the correlation values were still relatively weak to moderate.

Furthermore, when associated with the results in Figure 2, it was found that some features that were previously identified as containing placeholder values actually showed relatively high correlation values to the target variables, such as masa_corporal, segundamedicion_cintura, and several other anthropometric variables. This condition indicates the potential for statistical distortion, where non-clinical placeholder values can contribute to the formation of statistical relationships with the target variables.

This phenomenon shows that the existence of placeholder values not only affects data quality but can also affect the correlation structure between variables. The high correlation between features containing substitution values indicates the possibility of a pattern of system-generated encoding or sentinel values that are indirectly read as numerical signals by data-based models.

Thus, the integration of the results in Figure 2 and 3 reinforces the indication that the dataset has the characteristics of a systematic data encoding artifact, where invalid substitute values also affect the statistical distribution and the relationships between variables. This condition is an important concern because it can cause bias in the model learning process, especially in tree-based algorithms such as LightGBM, which are sensitive to extreme numerical value separation and have the potential to produce spurious decision patterns.

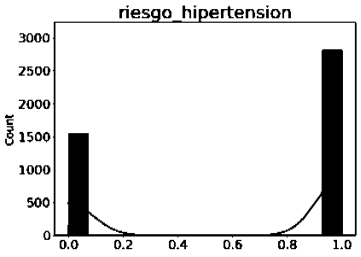

Figure 4. Class riesgo_hipertension Distribution

Furthermore, the distribution of class riesgo_hipertension is shown in figure 4, showing the class imbalance, where the Risky class (1) amounted to 2,816 data points and the Non-Risky class (0) amounted to 1,547 data points out of a total of 4,363 data points. This imbalance has the potential to cause bias in the model, as the model tends to recognize patterns more easily in the majority of classes.

3.3 Data Preparation

The aim was to prepare data for use in the modeling process [18]. This process includes the removal of irrelevant features (FOLIO_I = id), separation of data into features (X) and labels (y) with the risk of hypertension as the target, and division of data into training and test data with a 70:30 ratio using a train-test split.

In this study, data preparation not only focused on the technical aspects of modeling but was also designed to evaluate the impact of data quality on the clinical validity and robustness of the LightGBM model. Therefore, three data preprocessing schemes were designed to represent data conditions in real-world clinical data conditions as follows.

1) Scheme A – Original Data (Baseline / Observational Data)

This scheme uses the dataset in its original form without modification (as is) as obtained from the Kaggle public repository. The data in this scheme represent the condition of secondary observational data commonly found in electronic health records (EHRs) that have not undergone an advanced clinical validation.

In this condition, the dataset still contains invalid values or placeholder values that are used as system codes to represent data that are not available or cannot be measured. The value was still treated as numerical data by the algorithm. This scheme was used as a baseline condition to evaluate the sensitivity of the model to the presence of substitution values that have no clinical significance.

2) Scheme B – Cleaned Data (Static Cleaning / Missing Representation)

In this scheme, all identified placeholder values are converted to missing values (NaN) without imputation. This approach represents a more realistic clinical condition, in which not all of the patient's medical parameters are always available in the health record. In medical practice, data incompleteness is a common phenomenon that cannot always be replaced by estimates.

This scheme does not remove patient observations; therefore, the sample size is maintained. The main objective of this scheme was to evaluate the inherent ability of the LightGBM algorithm to handle sparse and incomplete data without additional value engineering.

3) Scheme C – Imputation Data (Clinical Reconstruction Approach)

This scheme is a more comprehensive approach to data reconstruction that combines statistical imputation methods and clinical consistency. The values that were converted to NaN in Scheme B were estimated using the K-Nearest Neighbors (KNN) imputer method, which utilizes the proximity of the characteristics between observations to estimate the missing values.

However, to maintain clinical validity, not all features were treated independently. Especially for the masa_corporal variable (BMI), the value is not imputed directly, but is recalculated deterministically based on the constituent variables, namely weight (peso) and height (stature), using the following formula:

$$BMI = \frac{Weight\ (kg)}{(Height\ (m))^2} \quad (9)$$

This approach was used to maintain physiological consistency between variables because BMI is a mathematical derivative that is not independent. Separate imputation of these variables has the potential to produce clinical inconsistencies, such as an unrealistic combination of weight and height in the resulting BMI value.

This scheme represents a clinical reconstruction scenario in which incomplete patient data are reconstructed using a statistical approach while maintaining biological limitations.

The flow of the stages of the preprocessing strategy is presented in the form of a diagram in Figure 5.

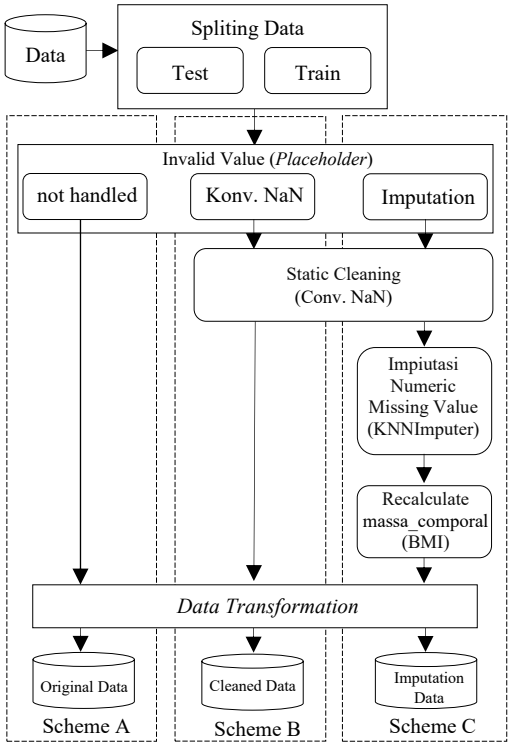

Figure 5. Data Preprocessing Strategy

The comparison of the three schemes is not only aimed at evaluating the performance of the model but also at simulating

Corresponding Author: mochandrisetiawan2@gmail.com
Doi: 10.30996/elsains.v8i1.133544

three different conditions in the context of real-world clinical data:

1) Scheme A (Original Data) represents raw observational data that have not been validated, including potential logging errors and internal system codes.
2) Scheme B (Cleaned Data) represents the real-world condition of incomplete medical records, where patient data are not always complete and not engineered.
3) Scheme C (Imputation Data) represents the condition of reconstruction of clinical data based on statistical estimation by maintaining biological consistency between variables.

Thus, this approach allows the evaluation of the model not only from the perspective of prediction accuracy but also from the perspective of robustness, clinical plausibility, and data realism sensitivity.

The data transformation results of these three data preprocessing schemes are presented in Figure 6, 7, and 8, respectively, to evaluate the comparison of data conditions before being input into the LightGBM algorithm.

0	1	2	3	4	0	4358	4359	4360	4361	4362
74.55	222.22	222.22	81.5	222.22	...	222.22	56.25	222.22	222.22	69.15
147.3	151.0	151.0	173.4	151.0	...	151.0	168.4	151.0	151.0	161.4
222.2	222.2	222.2	98.5	222.2	...	222.2	80.0	222.2	222.2	98.0
64.7	96.75	68.7	64.7	97.15	...	64.25	64.7	71.2	68.2	64.7
154.0	152.2	144.8	154.0	161.3	...	160.5	154.0	148.6	167.7	154.0
48.5	44.5	42.3	48.5	49.6	...	51.1	48.5	47.2	51.5	48.5
33.5	41.1	37.8	33.5	42.0	...	34.5	33.5	38.0	36.5	33.5
222.2	113.7	103.7	222.2	118.9	...	77.8	222.2	222.2	96.4	222.2
107	104	105	117	123	...	125	130	147	108	100
4	2	1	5	2	...	4	4	4	2	4
32.889389	1.0	1.0	26.265339	1.0	...	1.0	19.193721	1.0	1.0	25.600777
120	240	480	275	255	...	280	780	300	240	300

Figure 6. Original Data

24	4322	1759	791	0	3863	1702	389	1607
63.65	103.0	86.35	42.4	...	80.0	56.25	127.5	63.8
183.8	159.8	164.0	146.9	...	161.5	151.3	171.2	158.5
77.6	120.7	106.6	67.1	...	97.2	82.2	132.8	87.6
64.7	64.7	64.7	64.7	...	64.7	64.7	64.7	64.7
154.0	154.0	154.0	154.0	...	154.0	154.0	154.0	154.0
48.5	48.5	48.5	48.5	...	48.5	48.5	48.5	48.5
33.5	33.5	33.5	33.5	...	33.5	33.5	33.5	33.5
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
114.0	NaN	NaN	112.0	...	146.0	102.0	112.0	104.0
3.0	2.0	5.0	4.0	...	2.0	3.0	2.0	4.0
18.328214	38.852821	30.990106	18.822291	...	29.576568	23.585607	42.099095	24.461905
660.0	300.0	150.0	600.0	...	80.0	150.0	600.0	240.0

Figure 7. Cleaned Data

0	1	2	3	0	3050	3051	3052	3053
63.65	103.0	86.35	42.4	...	80.0	56.25	127.5	63.8
183.8	159.8	164.0	146.9	...	161.5	151.3	171.2	158.5
77.6	120.7	106.6	67.1	...	97.2	82.2	132.8	87.6
64.7	64.7	64.7	64.7	...	64.7	64.7	64.7	64.7
154.0	154.0	154.0	154.0	...	154.0	154.0	154.0	154.0
48.5	48.5	48.5	48.5	...	48.5	48.5	48.5	48.5
33.5	33.5	33.5	33.5	...	33.5	33.5	33.5	33.5
102.96	101.42	101.66	91.66	...	101.02	98.56	98.5	96.86
114.0	122.6	134.8	112.0	...	146.0	102.0	112.0	104.0
3.0	2.0	5.0	4.0	...	2.0	3.0	2.0	4.0
18.328214	38.852821	30.990106	18.822291	...	29.576568	23.585607	42.099095	24.461905
660.0	300.0	150.0	600.0	...	80.0	150.0	600.0	240.0

Figure 8. Date Imputation

Next, a transformation is performed on categorical features, namely sexo or gender. Although it has been represented in numerical form (1 for male and 2 for female), the value is still categorical. Therefore, encoding was performed using OneHotEncoder to generate binary representations that are more suitable for modeling [13].

Dataset Awal	Mapping Data	OneHotEncoder
1	Male	1
2	Female	0

Figure 9. Categorical transformation

Based on Figure 9, in the initial dataset, the sex variable is still in the form of numerical categories, namely, 1 for male and 2 for female. Next, a mapping process was carried out, where the value of 1 was changed to Male and the value of 2 was changed to Female. Subsequently, One-Hot Encoding was performed to convert categorical data into binary representations.

Normalization was performed on all numerical features to equalize the scale of the values between the variables. Normalization was performed using MinMaxScaler by converting the values into a range of 0 to 1, so that each feature had a balanced contribution and did not dominate each other [24], as shown in Figure 10.

```
array([[0.46590909, 0.58992806, 0.59375   , ..., 0.75      , 0.27199684,
        0.08079268],
       [0.46590909, 0.58992806, 0.59375   , ..., 0.5       , 0.25697455,
        0.02591463],
       [0.36363636, 0.58992806, 0.59375   , ..., 0.5       , 0.2157569 ,
        0.04420732],
       ...,
```

Figure 10. Normalization Results

The entire process is applied in a preprocessing pipeline, where the imputation is only fitted to the data training and transformed to the data testing to avoid data leakage. As a result, the data become cleaner and more consistent without any relearning process [10].

3.4 Modeling

At this stage, the selection and setting of model parameters are performed to obtain optimal results [18]. This study used a LightGBM-based method based on previous studies, with an accuracy of 99.69%. After the data were prepared, several experimental schemes were carried out to evaluate the effect of data quality on model performance, by way the model was trained and tested using three data conditions, namely original, clean, and imputation data.

The class imbalance is overcome by assigning greater weight to the minority class and less weight to the majority class during training. In LightGBM, it is implemented through the parameter class_weight = "balanced" which adjusts the proportion of the class in the loss function. This approach increases the sensitivity of the model to minority classes [25].

RandomizedSearchCV is a random sampling-based hyperparameter tuning method that searches for optimal combinations of parameters and evaluates them using cross-validation. This method is computationally efficient and effective for complex models such as LightGBM [26]. The experimental scheme includes the use of hyperparameter value candidates, as presented in Table 3.

Table 3. Random Search Hyperparameters

Hyperparameter	Candidate Values
n_estimators	[100, 200, 300, 400, 500]
learning_rate	[0.001, 0.01, 0.05, 0.1, 0.2]
max_depth	[3, 5, 7, 9, -1]
num_leaves	[15, 31, 63, 127]
min_child_samples	[10, 20, 30, 50, 100]
subsample	[0.6, 0.7, 0.8, 0.9, 1.0]
colsample_bytree	[0.6, 0.7, 0.8, 0.9, 1.0]
reg_alpha	[0, 0.1, 0.5, 1.0]
reg_lambda	[0, 0.1, 0.5, 1.0]

The process of searching for optimal hyperparameter combinations was carried out independently on each data scheme using Repeated Stratified 10-Fold Cross-Validation (n_repeats = 5) with the F1-score as the objective function. In this approach, the validation process was repeated five times with a different fold division for each repetition, resulting in a more stable and representative evaluation. In addition to allowing a more thorough exploration of the hyperparameter space, this approach also helps reduce the risk of overfitting certain data subsets and improves the generalization capabilities of the model. The results of the best combination of hyperparameters, along with the optimal F1-score estimation of the tuning process, are summarized in Table 4.

Table 4. Random Search Parameters

Hyperparameter	Data		
	Original	**Cleaned**	**Imputation**
n_estimators	200	300	300
learning_rate	0,05	0,2	0,2
max_depth	3	5	5

num_leaves	63	63	63
min_child_samples	10	50	50
subsample	0,7	0,6	0,6
colsample_bytree	1,0	0,9	0,9
reg_alpha	0	0,1	0,1
reg_lambda	0	0,5	0,5
Random SearchCV *Results*	0.9965	0.9964	0,9952

Based on Table 4, the results of RandomizedSearchCV show that the cleaned data scheme and imputation data produce the same best hyperparameter combination, but with different performance values, namely, the F1-score of 0.9964 in the cleaned data and 0.9952 in the imputation data. The similarity of these hyperparameters indicates that the optimization process tends to converge on the same configuration after handling placeholder values, while the difference in performance values indicates that the imputation method still affects the distribution of data and causes slight variation in the model performance. Furthermore, the best hyperparameter combination of RandomizedSearchCV was used to retrain the model on all training data to produce the final model used in the evaluation stage.

3.5 Evaluation

This evaluation aims to review the model development process and ensure its suitability for the business objectives that have been set [18]. The evaluation was performed in three stages.

1) Model evaluation using RandomizedSearchCV with Repeated Stratified 10-Fold Cross-Validation (n_repeats = 5).
2) The test data (30%) were evaluated using accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC.

The Deployment stage was not carried out because the research focused on model evaluation and had not yet arrived at the implementation of the system.

4 Discussion

The evaluation results for each LightGBM-based model trial scheme are as follows:

4.1 K-Fold Cross Validation Random SearchCV Validation Results

Table 5. RandomizedSearchCV K-Fold Cross Validation Results

Dataset	Accuracy ± Dev.
Original Data	0.9955 ± 0.0044
Cleaned Data	0.9954 ± 0.0044
Imputation Data	0.9938 ± 0.0043

Based on Table 5, model evaluation using RandomizedSearchCV integrated with Repeated Stratified 10-Fold Cross-Validation (n_repeats = 5) shows that the LightGBM model has high and consistent performance across all data preprocessing schemes. The evaluation results displayed in the form of average accuracy and standard deviation between folds show that in the original data scheme, an accuracy of 0.9955 ± 0.0044 was obtained, followed by clean data of 0.9954 ± 0.0044 and imputed data of 0.9938 ± 0.0043. The low standard deviation in all the schemes indicates that the model has a small variance and is stable across various data divisions.

The high consistency between the accuracy value and the low standard deviation indicates that the model not only produces high performance but is also stable across various data subsets. This reduces the indication of overfitting, which is usually characterized by large performance fluctuations between folds or reliance on a single split bias. Thus, the use of repeated cross-validation with five iterations reinforces the validity of the evaluation and shows that the model's performance reflects more robust generalization capabilities rather than simply the optimistic outcome of a single data-sharing scheme.

4.2 Classification Report

Table 6. Classification Report Scheme A : Original Data

Class	Precision	*Recall*	*F1-score*	*Support*
0	0.993562	0.997845	0.995699	464
1	0.998814	0.996450	0.997630	845
Accuracy			0.996944	1309

Based on Table 6, the LightGBM model on Scheme A (original data) yielded an accuracy of 99.69% with a very high F1-score in both classes. Although it shows excellent performance, this condition indicates a potential optimistic bias due to the existence of placeholder values that are still treated as valid numerical features. Under these conditions, the model has the potential to capture artificial patterns of extreme values or system codes, such that some class separations do not fully represent the actual clinical relationship but are rather influenced by unnatural data structures.

These findings provide an important basis for analyzing the impact of preprocessing on subsequent schemes, particularly in relation to how the handling of placeholder values can change the data distribution and reduce the model's dependence on these pseudo-patterns. Thus, the variation in performance between schemes reflects changes in accuracy and shows the level of robustness of the model to different data qualities as well as the extent to which the model learns from valid clinical signals.

Table 7. Classification Report Scheme B : Cleaned Data

Class	Precision	*Recall*	*F1-score*	*Support*
0	0.993521	0.991379	0.992449	464
1	0.995272	0.996450	0.995860	845
Accuracy			0.994652	1309

Furthermore, in Table 7, representing Scheme B (cleaned data), there was a relatively small decrease in performance of 0.23 percentage points, with an accuracy of 99.46%. This decrease suggests that when placeholder values are no longer treated as numerical features, the model loses some of the artificial separation patterns that previously contributed to the performance of Scheme A. Thus, these results indicate that some of the information used in the original data scheme is not entirely clinical but rather influenced by an unnatural data structure.

Nevertheless, the model's performance remained high and stable. This shows that LightGBM has a robust ability to handle missing values using a sparsity-aware split mechanism. When it finds a NaN value, LightGBM automatically allocates the sample to the best branch (node) that minimizes function loss based on the greatest information gain, so that the model can still form valid information signal-based decisions. This condition indicates that the elimination of the influence of placeholder values does not damage the model's predictive ability but rather improves the quality of learning by reducing the contribution of artificial patterns in the process of class separation.

Table 8. Classification Report Scheme C : Imputation Data

Class	Precision	*Recall*	*F1-score*	*Support*
0	0.993521	0.991379	0.992449	464
1	0.995272	0.996450	0.995860	845
Accuracy			0.994652	1309

In Table 8, which shows Scheme C (imputation data), the accuracy is recorded at 99.46%, which is numerically identical to the results in Scheme B (cleaned data). This similarity in performance indicates that the imputation process does not significantly change the data distribution or decision patterns of the model. This suggests that the missing information in Scheme B is not crucial to the main prediction structure.

The consistency of the results between Schemes B and C shows that the model has a high level of robustness to preprocessing variations, particularly in handling missing values.

Corresponding Author: mochandrisetiawan2@gmail.com
Doi: 10.30996/elsains.v8i1.133544

Thus, both the no-imputation and imputation approaches produced relatively similar learning patterns, indicating that the contribution of key information in the dataset remained stable despite value reconstruction.

4.3 Confusion Matrix

In addition, a confusion matrix was used to determine the distribution of the number of prediction results to the actual class and to identify the number of errors and the accuracy of the model's predictions.

Table 9. Confusion Matrix Results

Metric	Data		
	Original	Cleaned	Imputation
HCMC	842	842	842
TN	463	460	460
FP	3	3	3
FN	1	4	4
TP+TN	1305	1302	1302
FP+FN	4	7	7

Based on Table 9, it can be seen that across the schemes (original, cleaned, and imputation data), the model produces a consistent number of True Positives (TP) of 842, with a very low False Positive (FP) of 3 across all schemes. Meanwhile, small differences were observed in true negatives (TN) and false negatives (FN), where the original data scheme had the best performance with the lowest FN (1 case), while the other scheme showed FN in four cases.

Overall, the results in Table 6 show that the model has a very small error rate and can maintain predictive stability across the preprocessing scheme. The low FP and FN values indicate that the model is not only accurate overall but also consistent in classifying both classes well, which supports previous findings regarding model robustness.

4.4 ROC-AUC

Table 10. ROC-AUC Results on Each Data Scheme

Dataset	Accuracy ± Dev.
Original Data	0,9998928790
Cleaned Data	0,9998546215
Imputation Data	0.9998189145

Based on Table 10, the ROC-AUC evaluation results show that the model has a very high discriminating ability across all data preprocessing schemes, with an AUC value close to 1. The original data scheme produced the highest AUC value, followed by the cleaned and imputation data with very small but consistent declines.

Interestingly, although the classification report and confusion matrix values in Schemes B (cleaned data) and C (imputation data) showed identical results, there was a slight difference in the ROC-AUC values. This indicates that although the final prediction (threshold-based classification) produces the same performance, there is a small difference in the probability score distribution generated by the model for the two schemes.

These results show that the model has a stable performance in terms of both discrete and probabilistic classifications. The consistency in the confusion matrix and classification report, accompanied by small variations in the ROC-AUC, reinforces the indication that the model has high robustness to preprocessing variations in the data, without significant changes in class discrimination capabilities.

4.5 Robustness Discussion

Based on the results of the evaluation of the three data schemes, the LightGBM model showed a good level of robustness to changes in data quality conditions. This is shown by:

1) Repeated Stratified 10-Fold Cross-Validation results (n_repeats = 5) showed an average accuracy above 99% with a very low standard deviation (< 0.005) in all data schemes. These findings indicate that the model has a stable and consistent performance over various data divisions and does not show significant indications of overfitting.
2) The results of the classification report showed that the difference in accuracy between the schemes was relatively small, with an overall accuracy value above 99%. The precision, recall, and F1 scores in each class also remained high and consistent, indicating that the model was able to maintain stable performance and did not suffer from significant overfitting.
3) The results of the confusion matrix showed that the increase in the number of classification errors in each scheme was relatively small and insignificant, such that the model remained stable in making predictions under various data conditions.
4) The AUC value in all schemes was above 0.999, which signifies excellent classification ability in distinguishing between hypertensive and non-hypertensive classes.

5 Conclusion

Based on the results of the study, the LightGBM model showed very high performance in predicting hypertension risk in all test schemes, with an accuracy of 99.69% on the original data, while on the cleaned and imputation data, the same accuracy was obtained, which was 99.46%. This study also shows that data quality analysis and preprocessing strategies influence model performance, especially in handling invalid values in data. Despite a slight decrease in performance after the cleaning and imputation processes, the evaluation results of all schemes still showed stable performance. This is supported by the results of the classification report and confusion matrix, which show a relatively small difference in performance between the schemes, as well as an AUC value above 0.999 in all schemes, indicating that the model has a level of robustness and good generalization ability, and does not experience significant overfitting. In addition, the application of hyperparameter tuning using RandomizedSearch and class weight adjustment also supports the stability of the model performance. Machine learning pipelines and correlation analyses also showed that high model performance did not indicate data leakage or features that directly represented target-related features. However, the results of this study are still limited to evaluation using a single dataset; therefore, they cannot be generalized directly to a broader clinical condition.

For further research, it is recommended to test more diverse datasets and perform external validation to ensure the generalization capabilities of the model in real-world scenarios. In addition, exploration of more complex preprocessing and imputation methods can be performed to improve the balance between model performance and data representation validity.

Author Contribution

In this study, Moch. Andri Setiawan, Muhammad Najibulloh Muzaki, and Aidina Ristyawan have their respective roles in the implementation of the research. Moch. Andri Setiawan contributes to research design, data collection and processing, implementation of data mining methods, model testing, result analysis, and preparation of research manuscripts. Muhammad Najibulloh Muzaki and Aidina Ristyawan play a role in supervising research, validating results, and providing direction in the preparation of methodology and improvement of manuscript writing. All parties have read and approved the final manuscript.

Acknowledgment

The author also thanks all parties who provided guidance, direction, and support during the research process until it was completed.

Reference

[1] Mayo Clinic, "High blood pressure (hypertension)." [Online]. Available: https://www.mayoclinic.org/diseases-conditions/high-blood-pressure/symptoms-causes/syc-20373410. [Accessed: November 5, 2025].
[2] World Health Organization, "Hypertension." [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/hypertension. [Accessed: Nov. 5, 2025].
[3] A. Salshabilla, C. Y. Pardilawati, P. R. A. Sangging, and R. Z. Oktarlina, "Pathophysiology and Strategies

for Management of Hypertension Without Complications," *Marrow*, vol. 14, no. 4, pp. 640–646, 2024.

[4] I. Anita Silvianah, "THE RELATIONSHIP BETWEEN HYPERTENSION MEDICATION ADHERENCE TO BLOOD PRESSURE CHANGES IN THE ELDERLY AT THE ELDERLY POSYANDU," *J. Nursing*, pp. 52–61, 2024.

[5] A. Ristyawan, A. Nugroho, and T. K. Amarya, "Optimization of Random Forest Model Preprocessing for Stroke Prediction," *J. Tek. Inform. and Sist. Inf.*, vol. 12, no. 1, 2025.

[6] D. A. Murtiningsih, B. W. Sari, and I. N. Fajri, "Comparison of Light Gradient Boosting Machine , eXtreme Gradient Boosting , and CatBoost with Balancing and Hyperparameter Tuning for Hypertension Risk Prediction on Clinical Dataset," *J. Appl. Informatics Comput*. 9, no. 5, pp. 2753–2763, 2025.

[7] A. A. Aouragh, M. Bahaj, and F. Toufik, "Enhancing hypertension prediction : a hybrid machine learning optimization approach," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 37, no. 1, pp. 347–355, 2025, doi: 10.11591/ijeecs.v37.i1.pp347-355.

[8] A. Jarmakovica, "Machine learning-based strategies for improving healthcare data quality: an evaluation of accuracy, completeness, and reusability," *Front. Artif. Intell.*, vol. 8, no. July, pp. 1–14, 2025, doi: 10.3389/frai.2025.1621514.

[9] C. You, H. Dai, Y. Min, J. Sekhon, Joshi, and Singh S. Duncan, "Uncovering Memorization Effect in the Presence of Spurious Correlations," Nat. Commun., vol. 16, pp. 1–8, 2025, doi: 10.1038/s41467-025-61531-5.

[10] A. Ichwani, R. I. Kesuma, A. Setiawan, and I. E. Wicaksono, "Preventing Data Leakage in Classification via Integrated Machine Learning Pipelines: Preprocessing, Feature Transformation, and Hyperparameter Tuning," Journal of Information Technology (JUTIF), vol. 7, no. 1, pp. 391–410, Feb. 2026, doi: 10.52436/1.jutif.2026.7.2.5490.

[11] M. Reza Muzaki, I. Melinda Nurfajriana, F. Putri Anugerah Ilahi, A. Ristyawan, and E. Daniati, "Implementation of Data Mining with K-Nearest Neighbors Algorithm to Predict Diabetes Risk Using Telegram Chatbot," *August*, vol. 8, pp. 2549–7952, 2024.

[12] E. Daniati, R. Firliana, and A. S. W., Decision Support with Machine Learning Approach. Kediri, Indonesia: Universitas Nusantara PGRI Kediri, 2024.

[13] N. Kamilatutsaniya, E. Daniati, and M. N. Muzaki, "Modeling the Popularity Classification of Skincare Products Using a Support Vector Machine (SVM): A Comparative Study of Kernel Performance.," *Indonesian. J. Comput. Sci. Nothing.*, vol. 4, no. 2, pp. 86–95, 2025, [Online]. Available: https://subset.id/index.php/IJCSR/article/view/209

[14] C. Fan, M. Chen, X. Wang Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Ef fi cient and Reliable Knowledge Discovery From Building Operational Data," vol. 9, no. March, pp. 1–17, 2021, doi: 10.3389/fenrg.2021.652801.

[15] K. Hambardzumyan, "Data Preprocessing in Real-Time Education Management System," in Proceedings of the CSIT Conference 2021, Yerevan, Armenia, Sep. 27–Oct. 1, 2021, pp. 259–262.

[16] A. Apicella, F. Isgrò and R. Prevete, " do not Push the Button! Exploring Data Leakage Risks in Machine Learning and Transfer Learning," Artificial Intelligence Review, vol. 58, Art. no. 339, 2025, doi: 10.1007/s10462-025-11326-3.

[17] X. Hu *et al.*, "Blood pressure strati fi cation using photoplethysmography and light gradient boosting machine," *Front. Physiol.*, vol. 2030, no. February, pp. 1–11, 2023, doi: 10.3389/fphys.2023.1072273.

[18] A. Rianti, N. Wachid, A. Majid, and A. Fauzi, "CRISP-DM : Data Science Project Methodology," pp. 107–114, 2023.

[19] A. F. F. Jiménez and V. S. Sánchez Lee, "Hipertension Arterial Mexico Dataset." [Online]. Available: https://www.kaggle.com/datasets/frederickfelix/hipertensin-arterial-mxico/data. [Accessed: Oct. 22, 2025].

[20] M. A. Setiawan and M. H. Efendi, "The Effect of Preprocessing Techniques on the Performance of Explainable Boosting Machine (EBM) Models for Heart Attack Prediction, 8, no. 2, pp. 317–326, 2025.

[21] A. V. Chobanian, *et al.*, "The Seventh Report of the Joint National Committee on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure," *JAMA*, vol. 289, no. 19, pp. 2560–2572, 2003, doi: 10.1001/jama.289.19.2560.

[22] J. F. D. Min Ju Kim, Jung Hie Lee, "Circadian Rhythm Sleep Disorders," *J. Clin. Outcomes Manag.*, vol. 20, no. 11, pp. 513–528, 2014.

[23] World Health Organization, Physical Status: The Use and Interpretation of Anthropometry: Report of a WHO Expert Committee, WHO Technical Report Series No. 854. Geneva, Switzerland: World Health Organization; 1995.

[24] Elisa Tri Ammah A'Fena, Aidina Ristyawan, and Arie Nugroho, "The Effect of Feature Normalization on the Comparison of Random Forest Regressor and KNN Regressor Algorithms in Determining Smartphone Prices," *KRESNA J. Ris. and Service. Mass.*, vol. 5, no. 1, pp. 74–83, 2025, doi: 10.36080/kresna.v5i1.226.

[25] S. Khan, W. N. Ismail, and S. Alsalamah, "The Impact of Class Weight Optimization on Improving Machine Learning Outcomes in Identifying COVID-19 Specific ECG Patterns," vol. 2, no. Biostec, pp. 562–567, 2024, doi: 10.5220/0012413100003657.

[26] I. Print, "Comprehensive Analysis of Grid and Randomized Search on Dataset Performance, 3412, pp. 77–83, 2024, doi: 10.55581/ejeas.1581494.

Corresponding Author: mochandrisetiawan2@gmail.com
Doi: 10.30996/elsains.v8i1.133544