International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 16, No.1 (Aug-24)

http://dx.doi.org/10.12785/ijcds/160171

Classification of Tuberculosis Based on Chest X-Ray Images for Imbalance Data using SMOTE

Muhammad Fadhlullah Kh.TQ[1] **and Wahyono**[2]

[1]*Master Program in Artificial Intelligence, Universitas Gadjah Mada, Yogyakarta, Indonesia*
[2]*Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia*

Received 24 Oct. 2023, Revised 12 May 2024, Accepted 13 May 2024, Published 10 Aug. 2024

Abstract: This research delves into the challenge of dataset imbalance in classifying Chest X-Ray (CXR) images in the TBX11K dataset. To address this, the study employs Random Forest (RF) and XGBoost (XGB) methods, both with and without the Synthetic Minority Over-sampling Technique (SMOTE). The primary objective is to evaluate the impact of SMOTE on the performance of these models in classifying CXR images from the TBX11K dataset. This research applies SMOTE to the RF and XGB classification models to increase the number of minority class samples (TB positive) and address the imbalance with the majority class samples (TB negative). To ensure a comprehensive comparison, each model is assessed using a consistent set of evaluation metrics, including accuracy, precision, recall, and F1 score. The findings indicate that applying SMOTE to both RF and XGB models effectively mitigates class imbalance in the dataset. Specifically, the RF model without SMOTE achieves an accuracy of approximately 93.33%, while the RF model with SMOTE achieves an accuracy of 92.72%. On the other hand, the XGB model without SMOTE achieves an accuracy of 94.11%, and the XGB model with SMOTE reaches 94.33%. Although SMOTE enhances overall model performance, challenges persist in accurately predicting the minority classes 'altb' and 'ltb.' These challenges are attributed to the less representative features of these minority classes, which are difficult to overcome even with resampling techniques. Based on the experimental results, the XGB model with SMOTE emerges as the most optimal model for classifying TBX11K images. Despite the improved performance, further work is needed to enhance the prediction accuracy for minority classes, suggesting that additional techniques or more sophisticated models might be required to address this issue comprehensively.

Keywords: Random Forest, XGBoost, machine learning, VGG16

1. Introduction

A. Research Background

Tuberculosis (TB) is one of the leading causes of death worldwide [1]. According to data from the Ministry of Health of Indonesia in 2022, Indonesia ranks third after India and China in terms of the highest number of TB cases, with a total of 824,000 cases per year, out of which 93,000 result in fatalities. Early diagnosis plays a crucial role in improving the chances of recovery, preventing further spread, and significantly reducing mortality rates, in line with the WHO End TB Strategy [2] [3].

Chest X-ray (CXR) is widely regarded as the fastest and most cost-effective method for diagnosing TB compared to other diagnostic methods. However, the lack of skilled radiologists in TB detection reduces its effectiveness. Additionally, the high patient burden makes the diagnostic process less accurate. Therefore, a computer-based decision support system for TB detection using CXR is needed [4] [5].

The classification of TB in CXR images using machine learning faces the problem of imbalanced data, where the number of positive class (TB) samples is much lower than the negative class (non-TB) samples. If the data is imbalanced, the trained model may tend to predict more negative cases than positive, meaning more actual TB-positive patients go undetected. This can lead to incorrect treatment and wider disease spread. The Synthetic Minority Over-Sampling Technique (SMOTE) is one method to address imbalanced data by generating synthetic samples for the minority class [6]. In this case, using SMOTE can improve the accuracy of minority class classification by reducing the false-negative rate (FN).

Furthermore, to extract important features from CXR images, Convolutional Neural Networks (CNNs) can be used because of their ability to extract relevant features from images automatically. Convolutional Neural Networks (CNNs) have demonstrated success in numerous image-processing tasks, including the classification of TB based on

E-mail address: m.fadhlullah.khalilullah@mail.ugm.ac.id, wahyo@ugm.ac.id

Chest X-ray (CXR) images. In the study conducted by [7], TB detection involved combining CNN feature extraction with RF and XGBoost classification methods. The results showed that the combination of CNN for feature extraction and RF for classification achieved an accuracy of 98.667 and an AUC of 99.933 with a 90 training data percentage. However, this research was conducted on balanced CXR image data, with 6,000 data points consisting of 3,000 TB class and 3,000 normal class data points.

Therefore, this research will evaluate the effectiveness of SMOTE in addressing the data imbalance issue in TB CXR image data. CNN feature extraction and RF and XGBoost algorithms will be combined to create the classification model. Classification results will be compared between data without SMOTE resampling and data with SMOTE. The classification results to be compared include the F1-score as an appropriate evaluation method for imbalanced data.

B. Problem Statement

The problem statement in this research is that the chest X-ray image dataset TBX11K is experiencing class imbalance, as is typical in medical data, where the negative TB class is much larger than the positive TB class. This imbalance hinders classification models, which tend to prioritize learning from the majority class, consequently lowering their accuracy in predicting minority classes.

C. Research Limitations

In this study, several problem constraints are provided as follows:

1) The data used in this research is TBX11K obtained from Kaggle.
2) Image feature extraction is performed using a modified VGG16.
3) Classification is conducted using the Random Forest and XGBoost algorithms.
4) This research focuses on analyzing the impact of SMOTE on classification models with imbalanced image data.

D. Research Contributions

Based on the literature review conducted, previous research has addressed the issue of data imbalance by applying data sampling techniques, including the use of classification methods such as Random Forest and XGBoost, as explained in the studies presented in the literature review. However, there is no specific research that focuses on addressing the imbalance in Chest X-ray (CXR) image datasets by combining Random Forest or XGBoost classification methods with the SMOTE resampling technique.

2. Literature Review

Many previous studies have been conducted on TB classification using machine learning. For example [8], classified TB data into class 1 (positive) and class 2 (negative) using the Extreme Machine Learning (ELM) method. Through a confusion matrix evaluation, they achieved an accuracy of 99.33%, sensitivity of 99%, and specificity of 1, with the optimal number of hidden neurons being 20 and an optimal data split of 70:30 for training and testing data. However, these results were obtained from imbalanced and limited data, where class 1 consisted of 78 instances and class 2 had 22 instances.

In machine learning applications, it is common to encounter limited datasets or imbalanced classes, including in medical diagnosis data. In medical data, only 10% are diagnosed with a disease, while 90% are not. Most models trained on imbalanced data tend to have biases in predicting the majority class and neglecting the minority class.

One approach to tackle data imbalance is through data augmentation techniques, as demonstrated in the study by [9] on TB detection using CXR images. In their research, they analyzed 662 CXR images, consisting of 336 positive TB cases and 326 negative TB cases. Through various data augmentation methods, they increased the dataset to 2002 instances, with 1042 positive TB cases and 940 negative TB cases. Then, CXR features were inputted into different CNN architectures, and the results were fed into SVM for classification. Classification performance was evaluated using accuracy and AUC metrics. The classification results showed that data augmentation techniques like zooming, flipping, and rotating yielded similar results on MobileNet and VGG16 architectures. The best results were obtained from features extracted using MobileNet from the dataset augmented with rotation, achieving an accuracy of 96.6% and an AUC of 0.99. Although the results were excellent, this study used a limited dataset. Additionally, it is worth noting that the variations generated by augmentation techniques are pseudo-variations, meaning they do not reflect the true variations in the original data. This can lead to overly optimistic model evaluation results, especially if the variations generated by augmentation are much larger than the variations in the original data.

Therefore, other techniques need to be considered to address data imbalance while considering the data's variability characteristics, such as the SMOTE technique. For example, a study [10] compared the performance of an Artificial Neural Network (ANN) model with and without SMOTE for television advertisement rating classification. Experimental results showed that ANN's performance with SMOTE achieved an accuracy of 87.06%, compared to 86.35% without SMOTE. Another study [11] also investigated the impact of SMOTE, but with a different model, Random Forest (RF), to predict heart disease. The data consisted of 299 instances (96 patients and 203 non-patients) with an 80% training and 20% testing split, 12 independent variables, and 1 dependent variable. The results showed that SMOTE could reduce overfitting and improve RF model performance across all indicators. There was a 3.45% increase in Accuracy, 4.8% in Precision, 7.1% in Sensitivity, 4.8% in F1-score, 2.1% in Specificity, 4.4% in

G-Mean, and 6.3% in Youdens Index. In the same year, another study [12] also studied the impact of SMOTE on 8 classification models simultaneously. Their goal was to improve survival prediction performance for chronic heart failure patients. The eight machine-learning classification models included Random Forest (RF), Extra Tree (ET), Naive Bayes (NB), K-Nearest Neighbor (KNN), Decision Tree J48, Decision Table/Naive Bayes (DTNB), Optimized Forest, and Alternating Decision Tree (ADTree). The data were resampled with and without SMOTE for evaluation. The results showed increased accuracy across all classification models with SMOTE compared to without SMOTE, with DTNB as the classifier showing the highest increase of 5% (from 82.27% without SMOTE to 87.08% with SMOTE), becoming the model with the highest accuracy among the other classification models.

There are several other studies related to using SMOTE to address imbalanced data in various cases. Some classified imbalanced data using SMOTE and k-Nearest Neighbor (KNN) for Credit Card Fraud cases [13]. Others compared an enhanced model with other data resampling techniques, such as the study [14] comparing methods for detecting type 2 diabetes mellitus, using a Support Vector Machine (SVM) as the base model. The compared methods were ADASYN-SVM, SMOTE-SVM, and SVM without resampling. Through confusion matrix evaluation, they achieved an accuracy of 85.4% for SMOTE-SVM, 87.3% for ADASYN-SVM, and 83% for SVM. ADASYN-SVM emerged as the best method in this study, with a 4% increase in accuracy compared to the classifier model without resampling. In the same year, [15] compared the performance of several SMOTE types, including SMOTE, ADASYN (Adaptive Synthetic), SMOTE-TOMEK LINKS, and SMOTE-ENN (Edited Nearest Neighbor). They aimed to improve the diagnosis of cerebral tuberculosis using logistic regression (LR) and random forest (RF) models. The highest accuracy of 90.9% was achieved using SMOTE+TOMEK with LR modeling, showing a 6.81% improvement over the raw data (without resampling). Meanwhile, [16] introduced DeepSMOTE, an efficient transformative oversampling model combining SMOTE (Synthetic Minority Over Sampling Technique) with deep learning methods. DeepSMOTE meets three crucial characteristics as a good resampling algorithm: it operates on raw data, creates efficient low-dimensional embeddings, and generates high-quality synthetic images.

In the case of TB, [17] addressed data imbalance using SMOTE-Nominal (SMOTE-N) with a proportion of 13 for positive cases and 25 for negative cases, resulting in a balanced proportion of 25:26. While SMOTE-N reduced accuracy and recall levels compared to without SMOTE-N, it increased the f-measure, indicating that SMOTE-N was successful in balancing classes in pediatric TB data. This is because the f-measure reflects the classifier's goodness on the minority class. Additionally, the ROC Area value also increased, indicating better model performance and interpretation, as the classifier ranks randomly selected positive instances above randomly selected negative cases. This study concluded that SMOTE-N can improve machine learning performance on pediatric TB data with imbalanced classes.

For feature extraction purposes, Convolutional Neural Network (CNN) can be used due to its ability to extract relevant features from images automatically. A study [18] predicted hypertension and hyperlipidemia diseases using CNN for feature extraction. Their results showed that CNN-based feature extraction could reduce prediction bias and improve prediction effectiveness. However, the data used were text-based data from physical examination records. In the context of image data, a study [7] detected TB by comparing CNN feature extraction with RF and XGBoost classification models. They achieved an accuracy of 98.667% and an AUC of 99.933% using CNN-RF with a 90% training data percentage. However, this study was conducted on balanced CXR image data, with 3000 TB and 3000 normal data instances. Another study [19] also performed feature extraction on CXR images for COVID-19 identification using a hybrid convolutional neural network-principal component analysis (CNN-PCA). This technique was used to obtain discriminant features and reduce model complexity, where CNN extracted features from raw data which were then further extracted using PCA.

This study is centered around conducting experiments to assess the effects of SMOTE on the imbalanced class dataset of Chest X-ray (CXR) TBX11K. The dataset exhibits a notable class imbalance, with a considerably higher number of instances in the negative TB class compared to the positive TB class. This imbalance leads classification models to primarily identify the negative class. The study utilizes the CNN method with VGG16 architecture for image feature extraction, and for classification modeling, it employs Random Forest and XGBoost algorithms. The anticipated outcome of the study is to present and compare the model prediction results with and without utilizing SMOTE.

3. The Proposed Method

A. Research Description

The objective of this study is to evaluate the effectiveness of SMOTE in mitigating data imbalance during the classification of TB based on CXR images. Imbalance issues within CXR TB data can lead to classification errors, impacting model evaluation outcomes. To address this, the SMOTE resampling method, commonly used for handling imbalances in tabular data, will be applied to image data. To facilitate this process, a Convolutional Neural Network (CNN) with a slightly modified VGG16 architecture will be employed to extract features from CXR images. This will generate image features with dimensions suitable for processing during the SMOTE stage.

The choice of VGG16 as the feature extraction model in this study is based on findings from [20], which com-

pared ResNet50 and VGG16 and determined that VGG16 outperformed in classifying minority classes. The focus of this study on data balancing from feature extraction results drives this choice.

In terms of modeling, the Random Forest (RF) and XGBoost (XGB) classification algorithms will be utilized for classification tasks. The modeling process will include both with and without SMOTE resampling, resulting in four classification models: RF models with and without SMOTE, as well as XGBoost models with and without SMOTE. These models will be compared based on their performance using various evaluation metrics such as the F1-score.

B. Data Acquisition

The study utilizes the TBX11K dataset, publicly available on Kaggle [20], containing 11,200 CXR images annotated for TB classification. It includes six classes: Healthy (5,000 images), Sick non-TB (5,000 images, lung diseases other than TB such as pneumonia or bronchitis.), Active TB (924 images, symptomatic TB), Latent TB (212 images, asymptomatic TB), Active and Latent TB (54 images, symptomatic and latent), and Uncertain TB (10 images, challenging classification). The images have a resolution of 512×512 and serve as the basis for the research.

C. Algorithm Overview

As shown in Figure 1, the research model design begins with inputting the TBX11K CXR image dataset. The data first undergoes preprocessing to prepare it for input into the VGG16 architecture for feature extraction. Once the data is ready for processing, data splitting is performed to create a training dataset and a testing dataset. In both the training and testing phases, the data goes through the feature extraction process because the images will be represented by their features during the model creation (training phase) and prediction and evaluation processes (testing phase).

In the training phase, as this dataset experiences class imbalance, SMOTE resampling is applied to the training data to increase the number of samples from the minority class, approaching or even balancing it with the number of samples from the majority class. Subsequently, modeling is conducted using the RF and XGBoost algorithms for two different classification models. In the testing phase, the created models are evaluated using the testing dataset, and their performance is assessed using various evaluation metrics such as accuracy, precision, recall, and F1-Score.

D. Preprocessing Model

At this stage, each image data is prepared to be processed in the feature extraction stage using the VGG16 architecture. The preprocessing steps that will be performed include resizing the image data to 224×224 pixels, converting the image color space to RGB, and normalizing the pixel values of the image by subtracting the mean and dividing by the standard deviation for each color channel.

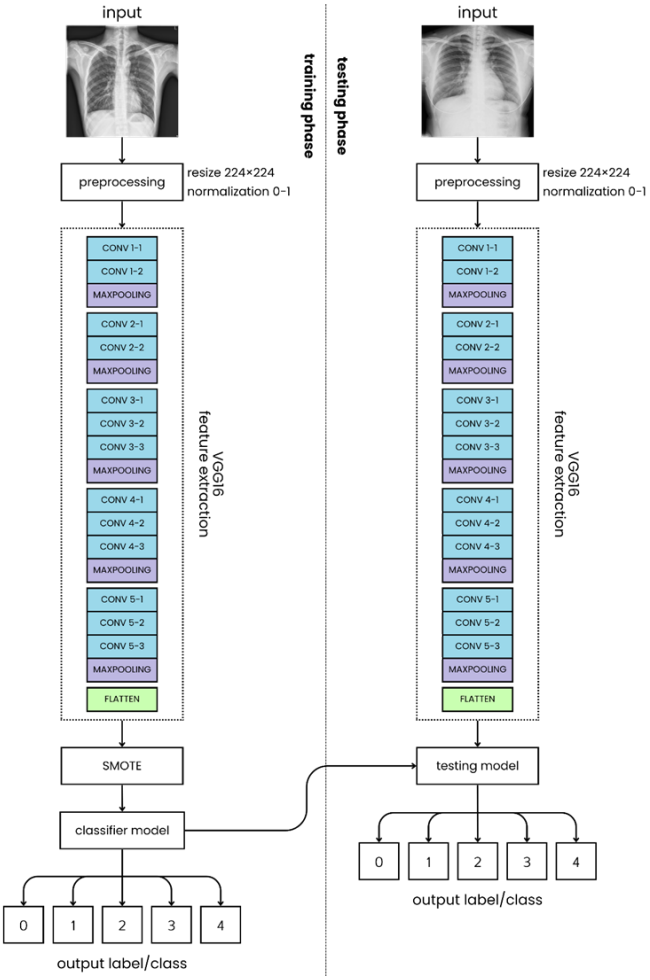

Figure 1. Research Model Algorithm

E. Feature Extraction

Feature extraction using the pre-trained CNN model with VGG16 architecture. VGG16 consists of 16 layers, where the first 13 layers are convolutional and pooling layers, while the last 3 layers are fully connected layers. However, in this study, VGG16 does not use fully connected layers and dropout because this model is only used for feature extraction, not for classification tasks. The classification task will use a separate machine-learning model. Therefore, the VGG16 model in this study is slightly modified by cutting off the last 3 layers and only using the 13 convolutional and max-pooling layers.

Additionally, this VGG16 model also uses pre-trained convolutional layers on the ImageNet dataset. This allows for transfer learning, as VGG16 can obtain weights from this pre-trained result. The convolutional layers in the VGG16 model are frozen because they are no longer trained but use the weights obtained from the pre-trained model. This allows VGG16 to save training time and computation only for weighting in the feature extraction phase. With this

TABLE I. Class and Data Split

Category	Class	Training	Testing	Total
Non-TB	healthy	3000	800	3800
	sick non-TB	3000	800	3800
TB	active TB	473	157	630
	latent TB	104	36	140
	active & latent TB	23	7	30
	TOTAL	6600	1800	8400

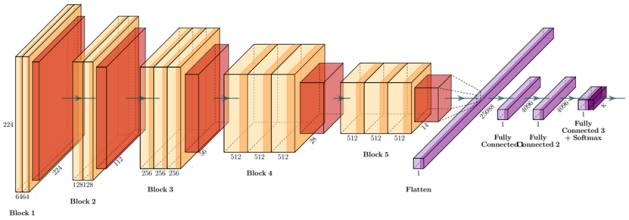

Figure 2. VGG16 Pre-trained CNN Architecture [21]

transfer learning, the trained VGG16 model can be used as a ”feature extractor” for CXR image recognition tasks without the need to train it from scratch.

VGG16 as a feature extractor is responsible for preparing image features to be SMOTE-ed and classified by the classifier. This is because to perform SMOTE, image features must be numeric vectors that can be computed for Euclidean distance. To convert image features obtained from extraction into a one-dimensional vector, it’s essential to perform a flattening operation. This operation essentially reshapes the multi-dimensional array of features into a single vector, allowing for further processing or input into machine-learning models. The flatten operation converts a matrix into a vector by arranging its elements sequentially. Thus, 25,088 image features will be produced that can be SMOTE-ed and classified by the classifier.

F. SMOTE

The SMOTE resampling technique will be applied to the training data. It’s important to note that SMOTE operates in the feature space, meaning its output consists of synthetic samples within the feature space, rather than synthetic data representing actual images. To apply SMOTE to image data, the image data has been previously prepared by VGG16 into one-dimensional vectors containing the image feature representations. SMOTE [11] will then create synthetic samples for the minority class by selecting several nearest neighbors for each sample point in the feature space and generating interpolations between these neighboring sample points. The interpolation process randomly chooses one of the neighbors and calculates the difference between the two samples. This difference is multiplied by a random number between 0 and 1 and added to the original minority sample. The result is a new sample that lies between the two minority samples. This way, SMOTE can increase the number of minority samples without eliminating variation in the image data.

Several hyperparameters can be modified when using SMOTE, including (1) ”sampling_strategy” which determines the ratio of the minority class to the majority class after the oversampling process. The default value is 1.0, which means the minority class will have the same number of samples as the majority class, and (2) ”k_neighbors” which specifies the number of nearest neighbors used to create synthetic samples. The default value is 5. Changing the value of ‘kneighbors‘ aims to obtain better and more accurate synthetic results.

SMOTE hyperparameters can influence the performance of the classification model trained with oversampled data. Hence, it is crucial to optimize these hyperparameters using k-fold cross-validation to ensure that the model used for predictions exhibits the best performance.

G. Classification Model

The classification will be performed using the RF and XGB algorithms, which leverage the features generated from VGG16 extraction. RF [22] employs ensemble learning techniques by combining multiple decision trees constructed randomly to improve classification accuracy. Each tree in the ensemble is built using a random subset of training data randomly sampled from the entire dataset. During classification, the model processes the input data’s features, and each tree in the ensemble generates class predictions. The final classification from the model is determined by taking the majority vote from all the trees in the ensemble. An illustration of the RF algorithm can be seen in Figure 3.

The advantages of RF include its ability to address overfitting because it divides the dataset into smaller subsets and makes decisions based on these subsets. Additionally, it can handle imbalanced data problems by using different sampling techniques.

XGB [25], on the other hand, utilizes an ensemble method that combines many gradient-boosted decision trees. In image data classification, the XGB algorithm can be employed to predict the class or label of each pixel or region within an image based on the features extracted from the image. An illustration of the XGB algorithm can be seen in Figure 4.

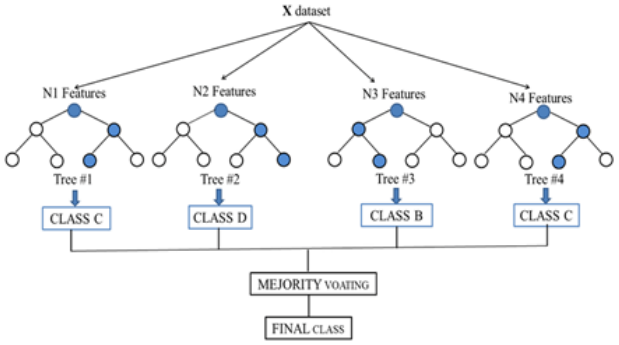

Figure 3. Illustration of Random Forest Classifier [23]

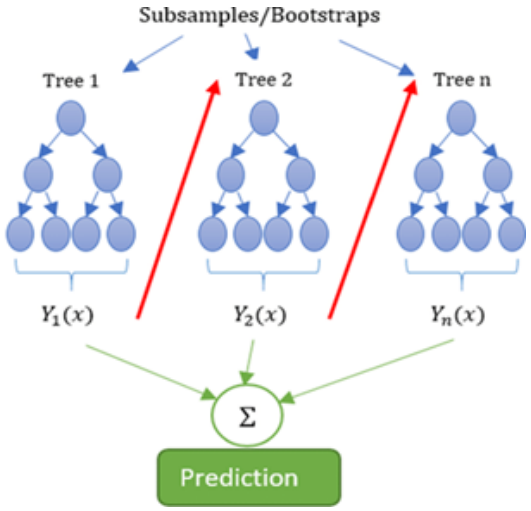

Figure 4. Illustration of XGBoost Classifier [24]

First, XGB initializes the model with a single decision tree that partitions the image data into several groups based on image features such as color, texture, shape, and more. This decision tree is referred to as the base tree or base learner. Next, XGB calculates the loss function of the base tree by comparing the predicted class to the actual class of the image data. The loss function indicates how well the model can distinguish between different classes within the image data.

Then, XGB attempts to reduce the loss function by adding new decision trees called additional learners. These additional trees are constructed in the same manner as the base tree but use image data that has been weighted based on the loss function of the base tree. Image data with higher loss functions receive larger weights, making them more influential in the formation of the additional trees. These additional trees are then combined with the base tree using summation operations to create a new and more accurate model.

Finally, XGB repeats these steps until it reaches the predefined number of decision trees or until there is no further improvement in accuracy. The final model produced by XGBoost is a combination of numerous decision trees that complement and strengthen each other.

4. Experiment and Results

A. Evaluation Strategy

In building the model, the training data is divided into an 80:20 split, meaning 80 is used for training, and 20 is reserved for model validation. This is done to assess the model's performance during the hyperparameter tuning process. After validation, the next step is to perform testing using a confusion matrix. The confusion matrix is used to measure how well the model can classify data into each class. For testing purposes, classification is also done with the model without SMOTE resampling to compare the results.

Model testing is carried out by splitting the dataset into two parts: training data and testing data. Training data is used to train the model, validation data is used to determine hyperparameters and prevent overfitting, and testing data is used to objectively evaluate the model's performance. Model evaluation is done by calculating accuracy, precision, recall, and F1-score. Accuracy measures how many correct predictions were made compared to the total number of data points. Precision measures how many correct positive predictions were made compared to the total number of positive predictions. Recall measures how many correct positive predictions were made compared to the total number of actual positive data points. F1-score is the harmonic mean of precision and recall, providing a balanced measure between the two.

B. Preprocessing and Feature Extraction Results

Data preprocessing was carried out to prepare the data for feature extraction. The preprocessing step employed was rescaling, where each pixel value in the images was transformed into a range between 0 and 1. This was achieved by dividing each original pixel value by 255. The purpose of this was to ensure that pixel values in the images had a consistent scale, thus making the model training process more stable and convergent. Additionally, the resolution of each image data was resized to 224×224 pixels to match the input resolution requirements of the VGG16 architecture. An example of resized image data is shown in Figure 5, representing image data from the Active and Latent TB classes.

In the feature extraction stage, both the training and testing data had their features extracted using VGG16 without the fully connected layer (FCL). This was done because the FCL layers are responsible for classifying data based on the extracted features from the preceding layers, whereas this study employed separate classification models using the RF and XGB classifier algorithms. As a result, an output dimension of (6600, 25088) was obtained, indicating that there were 6,600 image samples, with each image represented by 25,088 features. These features represent

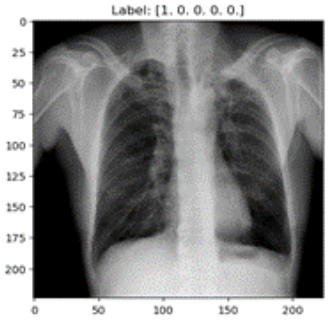

Figure 5. Example of Resized Image Data

```
with tf.device('/device:GPU:0'):
#Menghasilkan output dari model untuk data gambar
  features = model.predict(image_generator)

print("Output dimensi:", features.shape)
✓ 16m 52.6s
6600/6600 [==============================] - 1009s 153ms/step
Output dimensi: (6600, 25088)
```

Figure 6. Output Dimension of Feature Extraction Results

numerical representations of the images that have passed through the layers of the VGG16 model. This outcome is illustrated in Figure 6.

Although the extraction process is performed after data splitting, the VGG model used for feature extraction on both datasets is the same, thus not causing different information extraction as would occur if separate VGG models were used.

C. Model Training Results

In this stage, RF and XGB models were constructed both with and without SMOTE resampling to assess the impact of SMOTE on the performance of the RF model in predicting image data. Initially, model tuning was carried out to obtain optimal parameter configurations, and then the models were trained using these optimal configurations. The optimal configurations obtained were subsequently used as models for SMOTE tuning. The training results of models without and with SMOTE were then compared to observe the influence of SMOTE on the learning performance of the constructed models.

1) Training Results on Random Forest

Hyperparameter tuning was performed on the RF model with variations in the "n_estimator" parameter, controlling the number of decision trees to be built, and the "max_features" parameter, determining how many features should be considered when constructing each decision tree. In this experiment, "n_estimator (ne)" was set to three different values: 100, 500, and 1000, while "max_features" was varied at four different percentages: 25%, 50%, 75%, and 100% of the total available features, which were 25,088 features. The results of RF model tuning were evaluated based on several evaluation metrics, including accuracy, precision, recall, and F1-Score, summarized in Table II.

First, concerning the influence of the "n_estimator (ne)" parameter, as shown in Figure 7, the tuning results indicate that using 500 trees (n_estimator) leads to higher accuracy compared to 100 or 1000 trees, achieving an accuracy of approximately 93.41% when max_features=25% of the total number of features. In contrast, 100 and 1000 trees both reached an accuracy of 93.18%. This suggests that in the context of this dataset, using a larger number of trees does not always result in a significant improvement in model performance due to potential over-complexity.

Second, concerning the influence of the "max_features (mf)" parameter, it can be observed that the higher the percentage of features used, the model's performance slightly decreases, although not significantly based on the evaluation metrics, across different "n_estimator" values. For instance, in the case of 500 trees as shown in Figure 8, using 25% of the features results in the highest performance with an accuracy of 93.41%. Subsequently, the evaluation metrics indicate a slight decrease when 50% of the features are used, resulting in an accuracy of 93.03%. Further, the metrics show a slight decrease when 75% of the features are used, yielding an accuracy of 92.35%. The lowest performance is achieved when all features (100%) are used, with an accuracy of 92.12%. These results indicate that more features do not necessarily yield better outcomes, and, on the other hand, limiting the number of features used can help prevent overfitting and improve model performance.

Third, Table II shows that parameters that result in high accuracy also tend to yield good precision, recall, and F1-Score. This indicates that the model has good capabilities in classifying both positive and negative data and maintains a good balance between precision and recall.

Overall, the tuning results suggest that the Random Forest model with 500 trees and using around 25% of the features as max_features is the optimal configuration in the context of this dataset, achieving the best performance with an accuracy of approximately 93.41%. However, it's important to note that model performance does not always increase with parameter increments. There is a point where adding more estimators or maximum features no longer provides a significant performance improvement, as observed in some cases.

2) Training Results on Random Forest with SMOTE

The optimal RF configuration previously obtained, which is n_estimator=500 and max_features=25%, was applied in the SMOTE hyperparameter tuning process. The parameters optimized in this tuning stage were "sampling_strategy," controlling how many synthetic samples would be created for the minority class, and "k_neighbors," controlling the number of nearest neighbors to be considered when generating synthetic samples for the minority class. The evaluation was conducted on several combina-

TABLE II. Results of Random Forest Tuning

Parameter		Evaluation Metrics			
ne	mf	accuracy	precision	recall	F1-score
100	25%	93.18	90.97	93.18	91.91
100	50%	92.80	90.66	92.80	91.54
100	75%	92.73	90.58	92.73	91.56
100	100%	92.05	89.86	92.05	90.81
500	25%	93.41	91.24	93.41	92.16
500	50%	93.03	90.89	93.03	91.78
500	75%	92.35	90.20	92.35	91.11
500	100%	92.12	89.94	92.12	90.91
1000	25%	93.18	91.00	93.18	91.87
1000	50%	93.03	90.88	93.03	91.76
1000	75%	92.42	90.27	92.42	91.20
1000	100%	92.05	89.86	92.05	90.83

TABLE III. Results of SMOTE Tuning (Random Forest)

Parameter		Evaluation Metrics			
ss	kn	accuracy	precision	recall	F1-score
1500	3	97.95	97.96	97.95	97.95
1500	5	98.29	98.29	98.29	98.29
1500	7	97.86	97.86	97.86	97.85
2000	3	98.46	98.46	98.46	98.46
2000	5	98.83	98.84	98.83	98.83
2000	7	98.63	98.63	98.63	98.63
2500	3	98.44	98.45	98.44	98.44
2500	5	98.33	98.34	98.33	98.33
2500	7	98.56	98.56	98.56	98.55

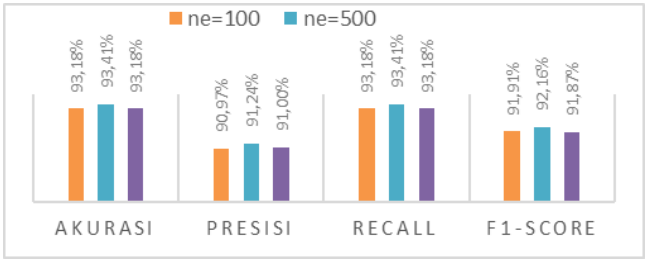

Figure 7. The Impact of ”n_estimator” (RF)

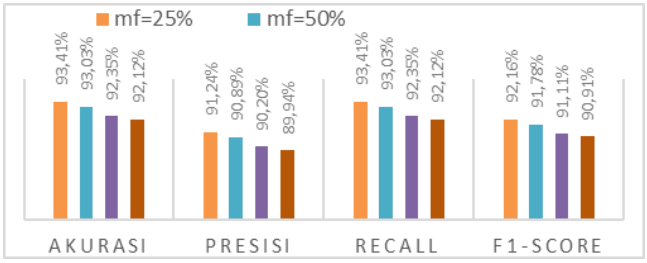

Figure 8. The Impact of ”max_features” (RF)

tions of ”sampling_strategy” and ”k_neighbors” values to assess the impact of parameter variations on model performance. The results are presented in Table III.

First, regarding the impact of the ”sampling_strategy” parameter, as shown in Figure 9 with k_neighbors=3, the tuning results indicate that a ”sampling_strategy” value of 2000 produces the best evaluation results with an accuracy of 98.458%, which is higher than 2500 samples with an accuracy of 98.444%. In contrast, 1500 samples resulted in the lowest performance with an accuracy of 97.952%. This suggests that increasing the number of synthetic samples generated by SMOTE can improve the model's accuracy. However, an overly aggressive ”sampling_strategy” value can lead to overfitting. Therefore, from this case, it can be stated that increasing the number of minority class data should be done judiciously and does not necessarily need to equal the majority class count.

Second, regarding the impact of the ”k_neighbors” parameter, as shown in Figure 10 with ”sampling_strategy”=2000, it can be observed that using 5 nearest neighbors results in the highest accuracy of around 98.833%, followed by 3 nearest neighbors at approximately 98.458%, and 7 nearest neighbors at around

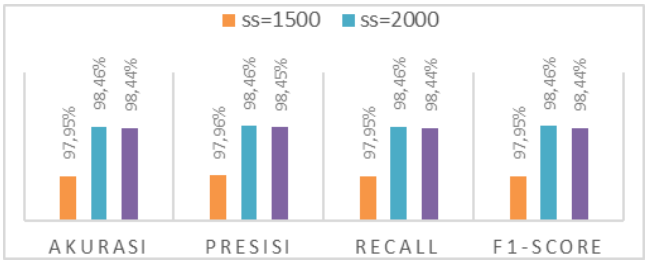

Figure 9. The Impact of ”sampling_strategy” (SMOTE RF)

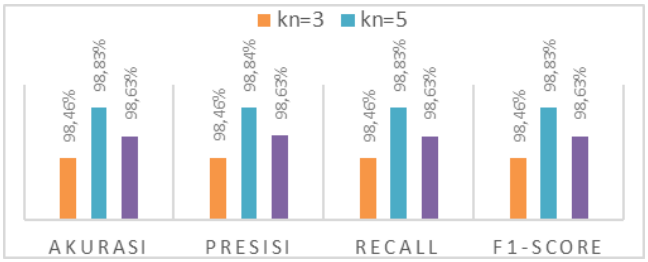

Figure 10. The Impact of ”k_neighbors” (SMOTE RF)

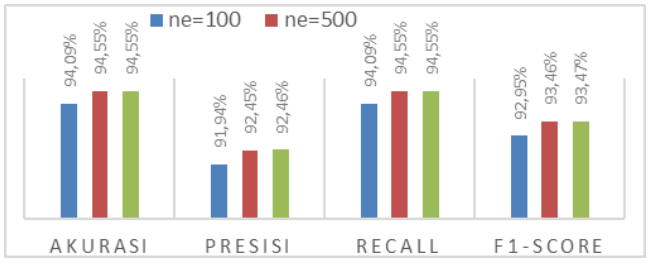

Figure 11. The Impact of ”n_estimator” (XGB)

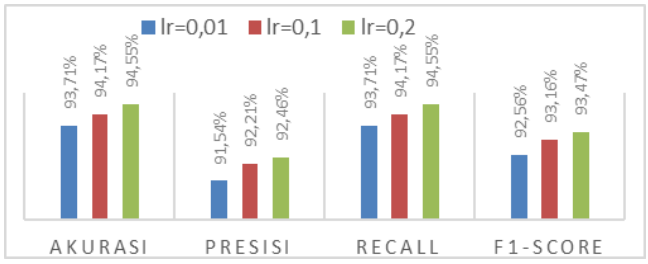

Figure 12. The Impact of ”learning_rate” (XGB)

98.625%. This suggests that employing a fewer or moderate number of nearest neighbors, such as 5 or 3, tends to yield better performance compared to using too many (e.g., 7). Therefore, it can be stated that a more moderate SMOTE approach can provide optimal results in boosting the minority class without causing overfitting.

Third, based on Table III, the optimal SMOTE configuration for the RF model for this dataset is ”sampling_strategy”=2000 and ”k_neighbors”=5. In the overall analysis, it can be concluded that using SMOTE with the optimal ”sampling_strategy” and ”k_neighbors” values can significantly enhance the performance of the RF model in handling class imbalance. This results in a model that can classify data more accurately, especially for the minority class.

3) Training Results on XGBoost

Hyperparameter tuning was performed on the second model using the XGB classifier algorithm with variations in the ”n_estimator” parameter, which controls the number of decision trees to be built, and the ”learning_rate” parameter, which governs the influence of each tree in the ensemble on the final model’s predictions. In this experiment, ”n_estimator” was set to three different values: 100, 500, and 1000, and ”learning_rate” was also varied with three different values: 0.01, 0.1, and 0.2. The tuning results of the XGB model were evaluated based on several evaluation metrics, including accuracy, precision, recall, and F1-Score, which are summarized in Table IV.

First, regarding the impact of ”n_estimator” on the XGB model, it can be observed that increasing the number of estimators generally improves the model’s performance, as shown in Figure 11, which refers to one of the tuning cases with ”learning_rate”=0.2. When ”n_estimator” increases from 100 to 1000, there is an improvement in all evaluation metrics. Accuracy increases from 94.091% to 94.545%, precision increases from 91.938% to 92.455%, recall increases from 94.091% to 94.545%, and F1-Score increases from 92.951% to 93.466%. From these results, it can be stated that using more estimators produces a stronger XGB model in classifying data.

Second, regarding the influence of the ”learning_rate” parameter, Figure 12 shows one of the tuning cases with ”n_estimator”=1000. It can be observed that overall, the model’s performance improves with increasing ”learning_rate” values from 0.01 to 0.2. This indicates that the model tends to achieve better results when it learns faster with a higher ”learning_rate” value. Therefore, based on these results, it can be stated that in this study, increasing the ”learning_rate” will enhance the model’s learning speed and improve its performance.

Based on the data presented in Table IV, the tuning results demonstrate a favorable balance between precision and recall in the XGB model. Even with variations in parameter settings, the precision and recall values consistently converge, indicating the model’s capability to identify the most positive instances while minimizing false positives accurately. Through this tuning process, the optimal configuration for the XGB model with 1000 trees and a ”learning_rate” of 0.2 is determined.

4) Training Results on XGBOOST with SMOTE

The optimal XGB configuration obtained earlier, which is ”n_estimator”=1000 and ”learning_rate”=0.2, is applied in the hyperparameter tuning process with SMOTE. The parameters optimized in this tuning phase are

TABLE IV. Results of XGBoost Tuning

Parameter		Evaluation Metrics			
ne	lr	accuracy	precision	recall	F1-score
100	0.01	92.50	90.51	92.50	91.40
100	0.1	93.56	91.36	93.56	92.39
100	0.2	94.09	91.94	94.09	92.95
500	0.01	93.64	91.47	93.64	92.49
500	0.1	94.24	92.28	94.24	93.22
500	0.2	94.55	92.45	94.55	93.46
1000	0.01	93.71	91.54	93.71	92.56
1000	0.1	94.17	92.21	94.17	93.16
1000	0.2	94.55	92.46	94.55	93.47

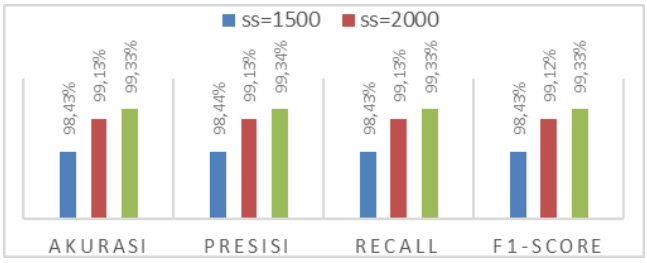

Figure 13. The Impact of ”sampling_strategy” (SMOTE XGB)

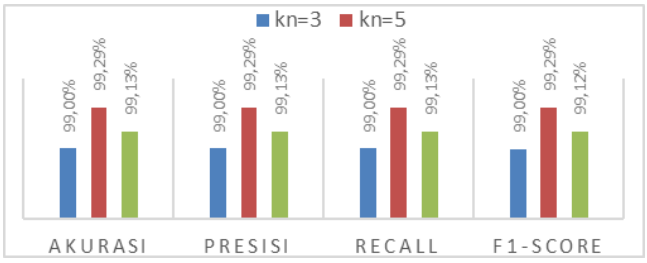

Figure 14. The Impact of ”k_neighbors” (SMOTE XGB)

”sampling_strategy,” which controls how many synthetic samples will be created for the minority class, and ”k_neighbors,” which controls the number of nearest neighbors to consider when creating synthetic samples for the minority class. Evaluation is carried out for various combinations of ”sampling_strategy” and ”k_neighbors” values to assess the impact of these parameter variations on the model’s performance. The results are shown in Table V.

First, regarding the impact of the ”sampling_strategy” parameter, as shown in Figure 13 for the case when ”k_neighbors”=7, the tuning results generally show improvement in each evaluation metric as the ”sampling_strategy” value increases. A higher ”sampling_strategy” value results in more synthetic samples being generated. This suggests that in the case of SMOTE with XGB, a more aggressive oversampling strategy can enhance the model’s ability to classify imbalanced data.

Second, regarding the influence of k_neighbors, Figure 14 shows a tuning case with a sampling_strategy of 2000. The graph indicates that the best evaluation results for each evaluation metric are obtained when k_neighbors is set to 5, rather than the highest value of ”k_neighbors”, although the differences in evaluation results are not significant. Therefore, it can be concluded that in the context of this research, the parameter k_neighbors should be set to an appropriate value, not too small or too large, to achieve optimal model performance and reduce the risk of overfitting.

Nonetheless, the tuning results indicate that all evaluation metrics have very high values and are close to each other. This indicates a good balance between the model’s ability to identify positive and negative instances and the minimization of false positives and false negatives.

In the overall analysis, this experiment reveals that an XGBoost model combined with SMOTE and optimal parameter configuration can produce a highly robust and reliable classification model for addressing class imbalance issues.

5) Discussion of Training Model Results

In the training model results stage, this study built two main models, namely RF and XGB, with and without using the SMOTE resampling technique to address the class imbalance in image data. The entire training model results are summarized in Table VI.

Next, in the RF model with SMOTE, the use of SMOTE with sampling_strategy=2000 and k_neighbors=5 resulted in the best performance with an accuracy of 98.83%. This indicates that SMOTE with a moderate strategy can significantly improve the model’s ability to handle class imbalance without overfitting.

Furthermore, in the XGB model without SMOTE, increasing the number of estimators (n_estimator) generally improved the model’s performance, with 1000 estimators and learning_rate=0.2 achieving the highest accuracy of 94.55%. Additionally, increasing the ”learning_rate” also improved the model’s performance. A good balance between precision and recall was observed in the XGB model. The optimal configuration for XGB was found to be 1000

TABLE V. Results of SMOTE Tuning (XGBoost)

Parameter		Evaluation Metrics			
ss	kn	accuracy	precision	recall	F1-score
1500	3	98.19	98.20	98.19	98.19
1500	5	98.29	98.30	98.29	98.29
1500	7	98.43	98.44	98.43	98.43
2000	3	99.00	99.00	99.00	99.00
2000	5	99.29	99.29	99.29	99.29
2000	7	99.13	99.13	99.13	99.12
2500	3	99.15	99.15	99.15	99.15
2500	5	99.11	99.11	99.11	99.11
2500	7	99.33	99.34	99.33	99.33

TABLE VI. Model Training Results

	Evaluation Metrics			
Model	accuracy	precision	recall	F1-score
RF	93,41	91,24	93,41	92,16
RF+SMOTE	98,83	98,84	98,83	98,83
XGB	94,55	92,46	94,55	93,47
XGB+SMOTE	99,33	99,34	99,33	99,33

estimators and learning_rate=0.2.

Finally, in the XGB model with SMOTE, tuning results showed that using SMOTE with sampling_strategy=2000 and k_neighbors=5 yielded the best performance with an accuracy of 99.29%. This indicates that the combination of XGB with SMOTE and optimal parameters produces a very strong and reliable model for addressing class imbalance.

Overall, both RF and XGB models can be configured with optimal parameters to address class imbalance. However, the XGB model, especially when combined with SMOTE, can achieve higher performance with very high accuracy. Therefore, in this case, the XGBoost model with SMOTE and optimized parameters is the best choice for predicting imbalanced class image data.

D. Results and Discussion on Testing Data

During the model testing phase, four different models were evaluated: RF without SMOTE, RF with SMOTE, XGB without SMOTE, and XGB with SMOTE, using the same evaluation metrics: accuracy, precision, recall, and F1 score. The summary of all model testing results is presented in Table VII.

The RF model without SMOTE achieved an accuracy of approximately 93.33% when tested on unseen data, indicating good generalization capabilities. Precision and recall also had a good balance, with an F1 score of around 92.02%. This model can be considered effective in classifying data with a high level of accuracy and a good balance between precision and recall.

On the other hand, the RF model enhanced with SMOTE produced a very high accuracy, reaching 92.72% on the test data, although slightly lower than the training data. Nevertheless, the precision and recall rates remained high, with an F1 score of about 91.59%. This model remained effective in classifying the test data and provided a high level of accuracy along with a good balance between precision and recall.

As for the XGB model without SMOTE, it achieved an accuracy of around 94.11% on the test data, indicating good generalization abilities. Precision and recall rates also remained high, with an F1 score of about 92.95%. This XGB model can be considered a strong choice for classification tasks on the dataset used.

Lastly, the XGB model enhanced with SMOTE achieved an accuracy of approximately 94.33% on the test data, although slightly lower than the training data. Precision and recall rates remained high, with an F1 score of around 93.27%. This model proved to be highly effective in addressing class imbalance issues and delivered strong performance on both the training and test data.

Overall, the testing results indicate that these models can perform well in classifying data, whether with or without SMOTE, with each model having its advantages and characteristics. Model selection depends on specific use-case goals and contexts, as well as the trade-offs between desired accuracy, precision, and recall for a particular application.

5. Conclusions and Future Work

Based on the results and discussion of the research, several conclusions can be drawn to address the research problem in this study. First, there are challenges in identify-

TABLE VII. Model Testing Results

Model	Evaluation Metrics			
	accuracy	precision	recall	F1-score
RF	93,33	90,86	93,33	92,02
RF+SMOTE	92,72	90,50	92,72	91,59
XGB	94,11	91,83	94,11	92,95
XGB+SMOTE	94,33	92,24	94,33	93,27

ing data in the "active&latent TB" minor class and latent TB class. Each RF and XGB model, whether with or without SMOTE, cannot accurately predict all data in these two classes. This is due to the less representative quality of features in these classes or biased feature selection. Despite the challenges as explained before, model prediction errors for all data from the "active&latent TB" class and latent TB class are still medically acceptable. This is because data in these two classes are predominantly predicted as the active TB class, which still falls under the positive TB category. Regardless of the challenges, the implementation of SMOTE on RF and XGB models improves model performance without experiencing overfitting. Results of the experiments show that SMOTE is effectively applied to the XGBoost model, where this model's performance is better than the XGBoost model without SMOTE. Meanwhile, the application of SMOTE is less effective on the Random Forest model, as seen from the model's performance not being better than the Random Forest model without SMOTE.

The study conducted has limitations and weaknesses in terms of variation, depth of the methods used, and the difficulty in predicting minority class data. Therefore, here are suggestions to continue the research:

1) In future studies, if using the TBX11K dataset, consider combining active TB, latent TB, and active & latent TB classes into a single class 'TB' as a representation of the general positive TB class. This is because the positive TB data alone is already limited, especially when divided into more specific classes. Additionally, chest X-rays are just one part of the TB disease identification process and not the key method for determining whether someone has TB or not.
2) Increase the variation of parameter values in the tuning process to observe the broader impact of parameters on the model. Experiment with model parameters and other SMOTE parameters not explored in this study.
3) Perform cross-validation to measure the training model performance after applying the SMOTE technique to obtain a more optimal model performance.
4) Consider improving the dataset quality or implementing other methods more suitable for handling the specific characteristics of the difficult-to-predict minority classes.
5) Explore the use of oversampling techniques other than SMOTE, such as ADASYN, or undersampling techniques to address class imbalance.
6) It is also important to consider the availability of additional data or collecting further data to enhance the model's capacity.

References

[1] S. Urooj, S. Suchitra, L. Krishnasamy, N. Sharma, and N. Pathak, "Stochastic learning-based artificial neural network model for an automatic tuberculosis detection system using chest x-ray images," *IEEE Access*, vol. 10, p. 103632–103643, 2022. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2022.3208882

[2] "Global tuberculosis report 2020: executive summary."

[3] S. V. G, N. Ponraj, and D. P. L, "Study on public chest x-ray data sets for lung disease classification," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*. IEEE, May 2021.

[4] U. Lopes and J. Valiati, "Pre-trained convolutional neural networks as feature extractors for tuberculosis detection," *Computers in Biology and Medicine*, vol. 89, p. 135–143, Oct. 2017. [Online]. Available: http://dx.doi.org/10.1016/j.compbiomed.2017.08.001

[5] R. Hooda, S. Sofat, S. Kaur, A. Mittal, and F. Meriaudeau, "Deep-learning: A potential method for tuberculosis detection using chest radiography," in *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. IEEE, Sep. 2017.

[6] A. Ali, S. M. Shamsuddin, and A. Ralescu, "Classification with class imbalance problem: A review," vol. 7, pp. 176–204, 01 2015.

[7] A. R. Laeli, Z. Rustam, and J. Pandelaki, "Tuberculosis detection based on chest x-rays using ensemble method with cnn feature extraction," in *2021 International Conference on Decision Aid Sciences and Application (DASA)*. IEEE, Dec. 2021.

[8] V. V. Nurdiansyah, I. Cholissodin, and P. P. Adikara, "Klasifikasi penyakit tuberkulosis (TB) menggunakan metode extreme learning machine (ELM)," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, vol. 4, no. 5, pp. 1387–1393, 2020.

[9] B. Oltu, S. Guney, B. Dengiz, and M. Agildere, "Automated tuberculosis detection using pre-trained CNN and SVM," in *2021 44th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, Jul. 2021.

[10] E. Sutoyo and M. A. Fadlurrahman, "Penerapan SMOTE untuk mengatasi imbalance class dalam klasifikasi television advertisement performance rating menggunakan artificial neural network," *J. Edukasi Dan Penelit. Inform. (JEPIN)*, vol. 6, no. 3, p. 379, Dec. 2020.

[11] E. Erlin, Y. Desnelita, N. Nasution, L. Suryati, and F. Zoromi, "Dampak smote terhadap kinerja random forest classifier berdasarkan data tidak seimbang," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 3, p. 677–690, Jul. 2022. [Online]. Available: http://dx.doi.org/10.30812/matrik.v21i3.1726

[12] S. Priyadarshinee and M. Panda, "Improving prediction of chronic heart failure using SMOTE and machine learning," in *2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA)*. IEEE, Sep. 2022.

[13] R. Siringoringo, "KLASIFIKASI DATA TIDAK SEIMBANG MENGGUNAKAN ALGORITMA SMOTE DAN k-NEAREST NEIGHBOR," *Journal Information System Development (ISD)*, vol. 3, no. 1, pp. 44–49, 2018.

[14] N. G. Ramadhan, "Comparative analysis of ADASYN-SVM and SMOTE-SVM methods on the detection of type 2 diabetes mellitus," *Sci. J. Inform.*, vol. 8, no. 2, pp. 276–282, Nov. 2021.

[15] K. Aftab, H. S. Fatima, N. Aziz, E. Baig, M. Khurram, F. Mubarak, and S. A. Enam, "Machine learning and sampling techniques to enhance radiological diagnosis of cerebral tuberculosis," in *2021 International Conference on Engineering and Emerging Technologies (ICEET)*. IEEE, Oct. 2021.

[16] D. Dablain, B. Krawczyk, and N. V. Chawla, "DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6390–6404, Sep. 2023.

[17] Y. E. Kurniawati, "Class imbalanced learning menggunakan algoritma synthetic minority over-sampling technique – nominal (SMOTE-N) pada dataset tuberculosis anak," *J. Buana Inform.*, vol. 10, no. 2, p. 134, Oct. 2019.

[18] S. Xie and H. Fan, "Research on CNN to feature extractionin diseases prediction," in *2019 International Conference on Computer Network, Electronic and Automation (ICCNEA)*. IEEE, Sep. 2019.

[19] M. Nahiduzzaman, M. O. F. Goni, M. S. Anower, M. R. Islam, M. Ahsan, J. Haider, S. Gurusamy, R. Hassan, and M. R. Islam, "A novel method for multivariant pneumonia classification based on hybrid CNN-PCA based feature extraction using extreme learning machine with CXR images," *IEEE Access*, vol. 9, pp. 147 512–147 526, 2021.

[20] Y. Liu, Y.-H. Wu, Y. Ban, H. Wang, and M.-M. Cheng, "Rethinking computer-aided tuberculosis diagnosis," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2643–2652.

[21] G. Vrbančič, M. Zorman, and V. Podgorelec, *Transfer Learning Tuning Utilizing Grey Wolf Optimizer for Identification of Brain Hemorrhage from Head CT Images*. University of Primorska Press, Oct. 2019, p. 61–66. [Online]. Available: http://dx.doi.org/10.26493/978-961-7055-82-5.61-66

[22] S. Misra and H. Li, *Noninvasive fracture characterization based on the classification of sonic wave travel times*. Elsevier, 2020, p. 243–287. [Online]. Available: http://dx.doi.org/10.1016/B978-0-12-817736-5.00009-0

[23] S. Saha and S. M. M. Ahsan, "Rice leaf disease recognition using gray-level co-occurrence matrix and statistical features," in *2021 5th International Conference on Electrical Information and Communication Technology (EICT)*. IEEE, Dec. 2021.

[24] C. EL Mazgualdi, T. Masrour, I. El Hassani, and A. Khdoudi, "Machine learning for kpis prediction: a case study of the overall equipment effectiveness within the automotive industry," *Soft Computing*, vol. 25, no. 4, p. 2891–2909, Oct. 2020. [Online]. Available: http://dx.doi.org/10.1007/s00500-020-05348-y

[25] H. Sunata, "Komparasi tujuh algoritma identifikasi fraud atm pada pt. bank central asia tbk," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 7, no. 3, p. 441–450, Dec. 2020. [Online]. Available: http://dx.doi.org/10.35957/jatisi.v7i3.471

Muhammad Fadhlullah Kh.TQ is freshly graduated from Universitas Gadjah Mada with a Master of Artificial Intelligence. He received a Bachelor of Science from Universitas Gadjah Mada in 2016. He is currently active as an Assistant Instructor at the Electronics and Instrumentation Laboratorium, Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia. Besides, he is also currently active as an Assistant Researcher at Geoseismal Research Center in role UI/UX and Front-End Engineer. His research interests include digital image processing, digital signal processing, machine learning, and software engineering.

Wahyono received a bachelor of computer science from Gadjah Mada University, Indonesia in 2010, and a doctoral degree at the Graduate School of Electrical Engineering, University of Ulsan, Ulsan, Korea. Since 2012, He has been serving as an assistant lecturer in the Department of Computer Science and Electronics, Faculty of Mathematics and Natural Science, Gadjah Mada University, Yogyakarta, Indonesia. He is currently an Associate Professor at the Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia. He is actively participating as a member of the societies as IEEE, ICROS. His research interests include digital image processing, pattern recognition, machine learning, computer vision, and software engineering. He has published many papers in reputable international journals indexed by Scopus in the fields of computer vision, image processing, and machine learning. He also served as a reviewer and editor in several international journals.