Journal of Future Artificial Intelligence and Technologies

E-ISSN: 3048-3719

Research Article

Integrating SMOTE-Tomek and Fusion Learning with XGBoost Meta-Learner for Robust Diabetes Recognition

De Rosal Ignatius Moses Setiadi [1,*], Kristiawan Nugroho [2], Ahmad Rofiqul Muslikh [3], Syahroni Wahyu Irianda [4], and Arnold Adimabua Ojugo [5]

[1] Department Informatic Engineering, Faculty of Computer Science, Dian Nuswantoro University, Semarang, Indonesia; e-mail : moses@dsn.dinus.ac.id
[2] Department of Information Technology and Industry, Stikubank University, Semarang, Indonesia; e-mail : kristiawan@edu.unisbank.ac.id
[3] Faculty of Information Technology, University of Merdeka Malang, Indonesia; e-mail : rofickachmad@unmer.ac.id
[4] Department of Informatics Engineering, Universitas Widya Gama Malang, Indonesia; e-mail : syahroni@widyagama.ac.id
[5] Department of Computer Science, Federal University of Petroleum Resources Effurun, Delta State, Nigeria; e-mail : ojugo.arnold@fupre.edu.ng
* Corresponding Author : De Rosal Ignatius Moses Setiadi

Abstract: This research aims to develop a robust diabetes classification method by integrating the Synthetic Minority Over-sampling Technique (SMOTE)-Tomek technique for data balancing and using a machine learning ensemble led by eXtreme Gradient Boosting (XGB) as a meta-learner. We propose an ensemble model that combines deep learning techniques such as Bidirectional Long Short-Term Memory (BiLSTM) and Bidirectional Gated Recurrent Units (BiGRU) with XGB classifier as the base learner. The data used included the Pima Indians Diabetes and Iraqi Society Diabetes datasets, which were processed by missing value handling, duplication, normalization, and the application of SMOTE-Tomek to resolve data imbalances. XGB, as a meta-learner, successfully improves the model's predictive ability by reducing bias and variance, resulting in more accurate and robust classification. The proposed ensemble model achieves perfect accuracy, precision, recall, specificity, and F1 score of 100% on all tested datasets. This method shows that combining ensemble learning techniques with a rigorous preprocessing approach can significantly improve diabetes classification performance.

Keywords: Diabetes Classification; Ensemble Learning; XGBoost Meta-Learner; SMOTE-Tomek; Deep Learning in Healthcare.

Received: April, 21st 2024
Revised: May, 17th 2024
Accepted: May, 22nd 2024
Published: May, 23rd 2024
Curr. Ver.: July, 3rd 2024

1. Introduction

Diabetes mellitus is a major challenge in global health, characterized by its chronic nature and significant contribution to morbidity and mortality worldwide[1], [2]. According to the World Health Organization (WHO), the prevalence of diabetes is expected to become the seventh leading cause of death by 2030[3]. These projections emphasize the critical need for early diagnosis and intervention, which can substantially reduce the serious complications associated with this disease. Current statistical data reveal an alarming increase in the prevalence of diabetes globally, almost doubling since 1980, caused by increasing cases of type 2 diabetes driven by obesity, aging, and unhealthy lifestyles[4], [5]. Early detection and accurate classification of diabetes can prevent many of these cases from progressing to serious complications such as nephropathy, retinopathy, and cardiovascular disease. Unfortunately, the majority of cases of Non-Communicable Diseases, including diabetes, are difficult to diagnose at an early stage, leading to under-treatment of the disease and significant reductions in health outcomes[4], [6].

In developing medical applications for early diabetes detection, various classification methods have been utilized to improve the level of prediction accuracy. Several widely used

diabetes datasets include Iraqi Society Diabetes (ISD) [7], Abelvikas[8], and PIMA Indians Diabetes (PID) [9], [10]. Each diabetes classification method developed will have a different performance in each database. Previous research was able to easily classify the Abelvikas dataset with an accuracy of up to 1.0 only with traditional machine learning methods[3], [11]. Datasets from ISD are still relatively easy to classify, because they can produce an accuracy of up to 0.99 in research[12]. While the PID dataset is the most popular and challenging diabetes dataset, many studies, such as [3], [4], [12]–[16], are only able to produce prediction accuracy of around 0.69 to 0.885. Even in research [3] when the same method could work with an accuracy of 1.0 on the Abelvikas dataset, it only produced an accuracy of 0.75 on the PID dataset. This shows that the input dataset significantly affects the method's performance. Furthermore, research in [17], which used the PID dataset produced a model with an accuracy performance of up to 0.93, and research in [1,18] achieved an accuracy of around 0.98 with a deep learning-based method. This shows the development of increasingly sophisticated classification methods. However, developing a classification model that is more adaptive and sensitive to intrinsic data variations in different diabetes datasets is urgent, in this case in order to obtain robust and accurate performance in various datasets.

Classification methods can generally be classified into three large groups: Machine Learning (ML), Deep Learning (DL), and Ensemble Methods. ML offers several approaches that have been used for a long time, such as decision trees, which are very easy to understand and interpret but are often prone to overfitting. Support Vector Machines (SVM) are very effective for high-dimensional data but are inefficient for large datasets because they tend to be slow[18]. Logistic Regression offers an easy-to-implement model and predicts results in the form of probabilities, but its performance suffers at complex and non-linear decision boundaries. DL, in this context through models based on Recurrent Neural Networks (RNN), provides unique capabilities in processing sequence or time series data, which is crucial for applications such as electronic medical records[19]. RNN, by default, can work well for temporal data but often experiences vanishing gradient problems. Other RNN methods, such as Long Short-Term Memory (LSTM) are more sophisticated because they are able to overcome this problem with gates that control the flow of information, making them better at learning long-term dependencies. Furthermore, there are also Gated Recurrent Units (GRU), which are newer, simplify the LSTM structure, and are usually more computationally efficient. However, both of these models require large datasets and long training times.

Ensemble methods can combine several ML or DL methods or both. These ML and DL methods are used as basic methods that produce initial predictions. Then, the final predictions are determined using several techniques, such as voting, stacking, or boosting. The more basic methods used can provide richer insights, but the complexity becomes more complex, and the computation becomes heavy. When using voting models, several models are applied independently, and more methods (can be more than three models) are generally used to produce maximum performance[20]–[23]. Voting tends to produce a more stable model that reduces the risk of overfitting through prediction aggregation. But it is sometimes less effective in dealing with the diversity and complexity of data because it only combines the final results of different models without considering the relationship between their predictions. Stacking involves training a secondary model, namely a meta-learner, to combine predictions from several base models. This allows the meta-learner to learn from mistakes made by the base model more flexibly than voting[24]. Boosting works by training models sequentially, where each new model tries to correct the errors made by the previous model[25]. This results in a series of models that focus on difficult cases that their predecessors failed to predict correctly, usually resulting in higher accuracy. One of the most famous boosting methods is eXtreme Gradient Boosting (XGB)[26].

Combining stacking and boosting can provide significant benefits because it combines the advantages of both techniques. Stacking allows us to combine models with different algorithms, including those that may tend to overfit or have weaknesses in certain aspects of the data. Meta-learners can learn how to combine these predictions most profitably. Boosting can effectively improve weak predictions made by individual models in the stack by focusing learning on difficult examples. This can reduce overall bias and variance while improving model generalization. This research combines these two methods, using XGB as a meta-learner from stacking. So, an ensemble model can be created that benefits not only from the collective wisdom of various learning algorithms but also from sequential learning that focuses on error reduction. This can produce very powerful models that take advantage of the

learning depth of the base model while gradually reducing errors through the boosting process.

In the context of medical data classification research, we often find relatively little and unbalanced data. This greatly affects the classification performance. This makes the process of balancing data with oversampling necessary. Because if undersampling is done, the data will become increasingly meaningless. Oversampling methods such as the synthetic minority over-sampling technique (SMOTE) are popularly used compared to random oversampling. Random oversampling methods generally do not provide significant or no effects[27]. One development of the SMOTE method is SMOTE-Tomek links[15]. This is a variant of SMOTE combined with the Tomek undersampling technique. In simple terms, SMOTE-Tomek links work by using SMOTE to add minority samples and then using Tomek Links to delete samples from the majority class that are too close to the minority class, thereby reducing overlap between classes. In this way, the dataset's quality improves, and ultimately, the machine becomes more effective when learning.

Based on the literature above, carrying out classification based on traditional ML methods is not possible to produce optimal results. The use of preprocessing methods such as balancing datasets using oversampling, feature selection, missing value imputation, or polynomial regression has the potential to improve performance. Deep learning methods also perform better in this case. So, this research proposes to combine several deep learning methods and SMOTE-Tomek sampling techniques in a stacking-boosting ensemble method for robust diabetes data recognition. Further contributions of this paper are:

1. Implementation of the SMOTE-Tomek sampling technique to improve distribution and quality.
2. Combining three methods, namely: BiLSTM and BiGRU, which are deep learning methods, and the XGB ensemble boosting method as a basic learning method for diabetes classification.
3. Combining three basic learning methods in a stacking-boosting ensemble, XGB is used, which is one of the boosting ensemble models used as a meta-learner in the stacking ensemble method.
4. Test the method on popular diabetes datasets to prove the method's robustness.

The next part of this paper will discuss preliminaries, which contain related works and important theories. Next, the proposed method, the results obtained from applying the method, and a discussion of the implications of these results in the broader context of diabetes classification are presented in detail. The discussion will include an in-depth analysis of the influence of the SMOTE-Tomek technique in balancing datasets, the effectiveness of ensemble models directed by the XGBoost meta-learner, how this combination improves the model's predictive ability over previous approaches, and ends with a conclusion.

2. Preliminaries

2.1 Bidirectional Long Short-Term Memory (BiLSTM)

BiLSTM is a DL model based on a Recurrent Neural Network (RNN). RNN is a neural network that handles data sequences, such as text or time series. RNNs have the feature that the output of the previous step is provided as input to the next step, helping the network to retain memory of previously processed information. Traditional RNNs experience the problem of vanishing gradient, where the gradient used in the learning process can become very small, making learning very slow or even stopping. To overcome this problem, LSTM, a variant of RNN, was developed. LSTM introduces the concept of gates, namely input gates (i_t), forget gates (f_t), and output gates (o_t) which effectively allows the network to learn when to "remember" and when to "forget" information that is no longer relevant. Apart from that, a cell state update function (C_t) was also added to help maintain relevant information over long data sequences without being affected by the vanishing gradient problem[28], [29]. Equation (1)-(4) shows the important formula for building the gates used.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{1}$$

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \tag{2}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3}$$

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \tag{4}$$

Where f_t is the activation of the forget gate at time t, σ is the sigmoid function, W_f is the weight of the forget gate, h_{t-1} is the hidden state from the previous timestep, x_t is the input at timestep t, b_f is biased for the forget gate, $\widetilde{C}_t$ is a candidate value for the memory cell, and h_t is the hidden state at time t.

BiLSTM is a further development of LSTM. BiLSTM processes data in two directions: forward and backward in separate layer formation. The first layer flows information from beginning to end, and the second layer flows information from end to beginning, so the network has context from the past and future at each point. This can improve the model's ability to understand both past and future context in the data sequence[30], [31]. This is especially useful for tasks such as language processing, where the context of the words before and after is very important. BiLSTM offers increased accuracy in classification and other complex tasks as well as flexibility in combining information from both directions. In terms of tuning, some key hyperparameters include the number of hidden units, which determines the complexity the model can handle; learning rate, which must be adjusted carefully to avoid slow or fast convergence; the number of layers, which affects the depth of the learning representation; dropout rate, to prevent overfitting; batch size, which affects the stability of gradient estimation and memory efficiency; and sequence length, which should be adjusted based on data context and task distribution. Proper setting of these hyperparameters is the key to optimizing BiLSTM performance.

2.2 Bidirectional Gated Recurrent Units (BiGRU)

GRU is a variation of RNN designed to overcome the same problem as LSTM, namely vanishing gradient, but with a simpler structure. GRU combines the input gate and forget gate into a single update gate, thereby reducing the number of parameters to be trained and speeding up the training process without sacrificing too much memory capability. BiGRU is a GRU implementation that processes data in two directions, similar to BiLSTM. By utilizing two GRU layers with two-way information flow, BiGRU is able to better capture the before and after context in data sequences[19], [32], [33]. GRU works with two types of gates, namely Update Gate and Reset Gate, which are explained in Equation (5) and (6), and new Hidden state, which is explained in Equation (7).

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{5}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{6}$$

$$\begin{aligned} \tilde{h}_t &= \tanh(W \cdot [r_t * h_{t-1}, x_t]) \\ h_t &= z_t * h_{t-1} + (1 - z_t) * \tilde{h}_t \end{aligned} \tag{7}$$

Where z_t is the update gate vector at time t, σ is the sigmoid function, W_z is the weight of the update gate, h_{t-1} is the hidden state from the previous timestep, x_t is the input at timestep t, r_t is the reset gate vector, W_r is the weight matrix for the reset gate, $\tilde{h}_t$ is the new hidden state candidate, W is the weight matrix, h_t is the updated hidden state.

The main hyperparameters in BiGRU include the number of hidden units, learning rate, number of layers, dropout rate, batch size, and sequence length. Effective tuning of these parameters requires experimentation and adjustments based on validation results to achieve a balance between training speed, memory requirements, and accuracy. BiGRU offers high computational efficiency because it has a simpler structure than BiLSTM. This model effectively understands bidirectional context, resulting in improvements in understanding dependencies in the data.

2.3 Boosting Ensemble

Boosting is an ensemble learning technique in machine learning that aims to create a robust model from a series of weaker models. This method works iteratively, where each newly added model attempts to correct the errors made by the previous model. Each model in this process focuses more on data samples that were difficult to predict by the previous model so that each subsequent model becomes more specific in overcoming the difficulties encountered[34], [35]. Boosting has a general working method as follows:

1. Initialization: Each data sample is given the same weight or weight based on distribution.
2. Iterative: trainer models are added one by one.
    a. The first model is trained on all the data.
    b. For each subsequent model, the data sample weights are adjusted so that the model focuses more on samples that the previous model had incorrectly predicted.
    c. This process is repeated until the maximum number of models is reached or additional models no longer improve accuracy.
3. Aggregation: the output of all models is taken by a certain method to get the final prediction.

While intuitively focusing on the wrong samples sounds like it will increase overfitting, boosting often shows good resistance to overfitting, especially if the number of models used is controlled. Boosting tends to be more effective in reducing bias and variance than other ensemble methods, such as bagging. Some popular boosting algorithms include Adaptive Boosting (AdaBoost), Gradient Boosting Machines (GBM), Extreme Gradient Boosting (XGBoost/XGB), LightGBM, and CatBoost. Where AdaBoost is the earliest boosting model, GBM is a boosting model that applies gradients from the loss function to guide the learning process, lightGBM is a lighter and faster version of GBM, XGB is an optimized version of GBM, and CatBoost is a boosting method that is more focused on getting high accuracy in data that has the majority of categorical data.

Regarding medical datasets with limited categoricality and require optimal performance, XGBoost was chosen in this research because it is a sophisticated implementation of current gradient-boosted trees. The main features of XGBoost include the addition of regularization to reduce overfitting, parallel processing that maximizes modern hardware, automatic handling of missing values, tree pruning with a depth-first approach, and integration of cross-validation which makes it easier to tune parameters efficiently[16], [36]. Important hyperparameters in XGBoost, such as max_depth, eta, subsample, and colsample_bytree, play an important role in optimizing the model. Tuning XGBoost involves adjusting these hyperparameters using techniques such as cross-validation and grid search to achieve a balance between training speed and accuracy.

2.4 Stacking Ensemble

Stacking is an ensemble machine-learning technique that combines predictions from multiple models to produce more accurate predictions. This method involves two levels of models: a first-level model, usually referred to as a base learner, and a second-level model that aggregates their predictions or is known as a meta-learner[2], [24]. In general, stacking works with several stages as follows:

1. Base Learners Training: different ML models are trained separately on the same dataset. These models can vary from linear regression, decision trees, SVM, neural networks, even DL models.
2. Predictions from Base Learners: each base learner makes predictions, which the meta-learner uses as input. These predictions can be class outputs or predicted probabilities.
3. Meta-Learner Training: the meta-learner is trained on predictions from the base learner as features with the same output/target as the initial training data. The goal is to learn how to combine base learner predictions best to improve accuracy.

The main advantages of stacking include diversifying the model through the use of multiple algorithms, which can reduce the risk of overfitting. In stacking, meta-learners such as linear regression or other ensemble models play an important role in integrating the output of the base learners, providing flexibility in combining predictions and increasing control over the integration process. The selection and complexity of meta-learners are critical, as they must be able to optimize and aggregate predictions effectively to minimize prediction error. The right meta-learner, especially one trained using out-of-fold predictions from the base

learner, can significantly improve model accuracy and robustness. Overall, the stacking technique harnesses the power of combining various models to produce very accurate and robust predictions.

2.5 Related Works

Various studies related to diabetes classification have been carried out, one of which is the research of Pradhan et al. [14], which tested several methods such as Naïve Bayes (NB), SVM, Random Forest, and Artificial Neural Networks (ANN) on the Pima Indian Diabetes (PID) dataset. The ANN model is structured with an input layer, several hidden layers, and an output layer, using Rectified Linear Unit (ReLU) and sigmoid activation functions to process data more effectively. This configuration allows the model to learn complex patterns in the data without being affected as much by overfitting or noise. Numerically, the test results show that the ANN achieves superior performance metrics compared to other models. Its accuracy reached 85.09% in the diabetes prediction task, surpassing other techniques significantly. Another study conducted by Wang et al. [13] introduced the DMP_MI algorithm. DMP_MI was designed to improve the accuracy of diabetes mellitus classification on the Pima Indians Diabetes (PID) dataset. The PID dataset has some problems with missing values and class imbalance. This algorithm uses the Naïve Bayes method to fill in missing values, the Adaptive Synthetic Sampling (ADASYN) method to balance classes in the dataset, and Random Forest as a classifier. Experimental results show that DMP_MI achieved an accuracy of 0.871, recall of 0.857, and precision of 0.806. This paper's conclusion confirms that combining data infill techniques, adaptive synthetic sampling, and robust classifiers can significantly overcome data quality problems and improve the effectiveness of medical diagnostic systems.

Özmen and Özcan's research [17] evaluated and compared four different approaches Classification and Regression Tree (CART), Artificial Neural Network (ANN), CART- Genetic Algorithm (CART- GA), and ANN-GA using the PID dataset. GA is assigned to adjust parameters in CART and ANN. Experimental results show that the CART-GA approach provides the best performance. Specifically, in testing using 10-fold cross-validation, the accuracy reached 93.42%, whereas without GA the accuracy was only 70.13%. In comparison, the traditional ANN model without GA optimization has lower accuracy, namely 59.74% in 10-fold cross-validation, whereas when applying GA, the accuracy is 81.82%. CART-GA consistently outperformed other approaches in all tested metrics—accuracy, precision, specificity, and F1 measure. This data shows that optimization using GA significantly increases the effectiveness of machine learning models in diagnosing Diabetes Mellitus.

Asniar et al.[15] proposed the Local Outlier Factor (LOF) method into SMOTE (SMOTE-LOF) for handling noise problems in imbalanced data. Keep in mind that most medical datasets are relatively unbalanced. The SMOTE-LOF method succeeded in increasing the accuracy of the classification model on various datasets, including the PID dataset, Haberman's Survival Data, and the Glass Identification Database. Specifically, for the PID dataset, SMOTE-LOF shows significant numerical accuracy improvements over both C4.5, NB and SVM. C.45 produced the best results with accuracy increasing from 71.09 with nothing to 73.03% with SMOTE to 75.13% and 75.10% for parameters k=3 and k=5 in SMOTE-LOF. These results confirm that both SMOTE and SMOTE-LOF sampling methods can effectively minimize noise's influence and improve predictive performance.

Chang et al. [4] compared several ML methods, such as NB, RF, and J48 decision trees to classify diabetes mellitus. The dataset used is also a PID dataset. Based on the test results, NB has good performance with feature selection, while RF is more effective when using more features. Apart from that, the best results were obtained by RF with an accuracy of up to 79.57%, a precision of 89.40%, and an AUC value of 86.24%. Similar research was conducted by Tasin et al. [16], here, the PID dataset is combined with several private datasets. Based on test results, using XGB and ADASYN, the accuracy reached 0.885, whereas, without ADASYN, the accuracy was only 0.78. This shows that the addition of synthetic data improves classifier performance.

Naz and Ahuja [37] used another approach in classifying diabetes on the same dataset, namely PID. A DL method with a multilayer perceptron artificial neural network and backpropagation technique is proposed to improve prediction performance. DL was tested and compared with other methods, such as ANN, NB, and DT. As a result, DL shows superior performance with an accuracy of 98.07%, which is much higher than other methods, which

have an accuracy of between 76% and 96%. With these results, DL is proven to be the most effective and promising method for use in early diagnosis of diabetes. A DL method called twice growth deep neural network (2GDNN) was also proposed by Olisah et al.[1]. This was done due to the limitations of previously used prediction methods that could not achieve the expected accuracy and problems with the PID dataset, such as missing values and non-normal data distribution. As a solution, innovative data processing methods are proposed, including the use of Spearman correlation for feature selection and polynomial regression for imputation of missing values. The 2GDNN method was also compared with the SVM and RF methods, and as a result, 2GDNN showed significant performance improvements, with accuracy, sensitivity, and F1 score all above 97%.

Previous research shows a variety of methods for diabetes classification, from traditional machine learning to deep learning and ensemble techniques. Although many achieve high accuracy, challenges remain in handling imbalanced datasets and data variations, especially in PIMA datasets. Therefore, this study proposes a combination of SMOTE-Tomek and stacking-boosting ensemble techniques with XGBoost as a meta-learner to improve diabetes classification performance.

3. Proposed Method

Inspired by various research that has been discussed previously and the theories described above. This research proposes a model that combines the BiLSTM, BiGRU, and XGB classifier methods as a base learner. The three base learners are combined using a stacking ensemble with an XGB regressor as a meta-learner. In addition, duplicate data, missing values, SMOTE-Tomek, and normalization were removed at the preprocessing stage. As an illustration, the proposed method is depicted in Figure 1.

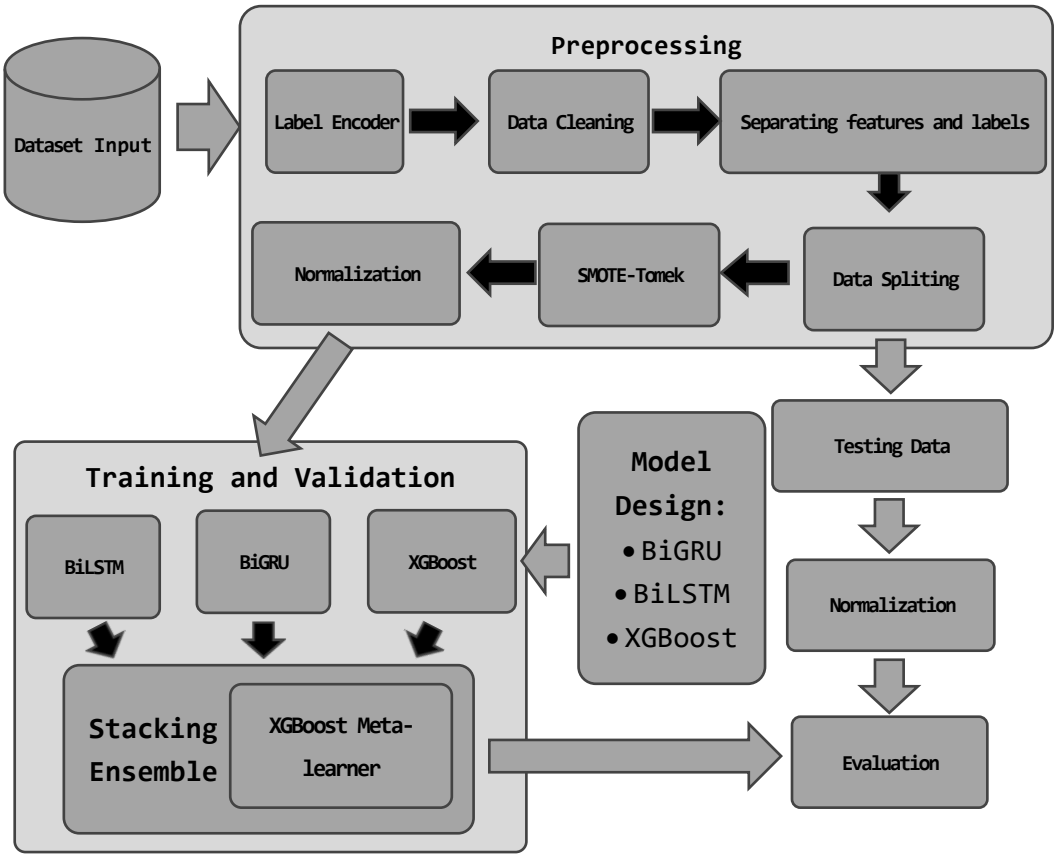

Figure 1. Proposed Method Illustration

Based on Figure 1, the stages are explained in more detail as follows:

1. Input data is read and stored in a data frame
2. Preprocessing is carried out in the following stages:

- Label encoder to convert non-numeric values into numeric values. For example, the feature 'Gender' with the values 'Male' and 'Female', will be converted to 0 and 1, so that the machine learning model easily processes it.
- Data cleaning is carried out by removing duplicates and missing values. Duplicate data was removed to avoid redundancy in the data. Meanwhile, deleting missing values on rows containing missing values is deleted to ensure data quality.
- Dividing Features and Targets: Dividing features and targets is done after data cleaning. The data needs to be separated as to what will be the input for the model (X) from what we want the model to predict (y).
- In the data-sharing process, the data is divided into training and testing sets with a ratio of 80:20.
- Class balancing on the training set was performed with SMOTE-Tomek. SMOTE is used for oversampling minority classes, while Tomek-Links is used for under-sampling the majority of classes, potentially creating overlapping classes.
- The final step in preprocessing is data normalization. Normalization was performed after SMOTE-Tomek because class balancing techniques may have changed the data distribution. Features are normalized with a standard scaler. Normalization using StandardScaler refers to the standardization process, which generally involves changing data features so that the distribution has a mean value of 0 and a standard deviation of 1. The mathematical formula for standardizing a feature is Equation (8).

$$z = \frac{(x - \mu)}{\sigma} \tag{8}$$

Where x is the original value of the feature, μ is the mean of the feature, σ is the standard deviation of the feature, and z is out standardization/normalization process.

3. Design and compile the three models, i.e.:
   - The BiGRU model design used is presented in Table 1.

Table 1. BiGRU Model Design.

No	Setting/Parameter	Value	Note
1.	RNN Layer	Bidirectional (GRU (64))	The first layer will have a bidirectional RNN layer with 64 GRU units.
		Bidirectional (GRU (32))	The bidirectional RNN layer with 32 GRU units is used for the second layer.
2.	return_sequences	True (for the first layer)	Returns the entire output sequence for the first layer.
3.	input_shape	(X_train_scaled.shape[1], 1)	The model expects sequences with a length according to the number of features in X_train_scaled and one feature per timestep.
4.	Dense layer	y_train_resampled.max() + 1	This layer has the same number of units as the classes in the y_train_resampled / output layer.
5.	Activation (Dense Layer)	softmax	The activation function is used in the output layer for multi-class classification.
6.	Optimizer	adam	Optimizer with a default learning rate value of 0.001.
7.	Loss Function	categorical_crossentry	Loss functions for multi-class classification
8.	Metrics	accuracy	Metrics that the model will evaluate during training and validation.

   - The BiLSTM model design used is presented in Table 2.

Table 2. BiLSTM Model Design.

No	Setting/Parameter	Value	Note
1.	RNN layer	Bidirectional(LSTM(64))	Bidirectional RNN layer with 64 LSTM units for the first layer.
		Bidirectional(LSTM (32))	The second layer will have a bidirectional RNN layer with 32 LSTM units.
2.	return_sequences	True (for the first layer)	Returns the entire output sequence for the first layer.
3.	input_shape	(X_train_scaled.shape[1], 1)	The model expects sequences with a length according to the number of features in X_train_scaled and one feature per timestep.
4.	Dense Layer	y_train_resampled.max() + 1	This layer has the same number of units as the classes in the y_train_resampled / output layer.
5.	Activation (Dense Layer)	softmax	The activation function is used in the output layer for multi-class classification.
6.	Optimizer	adam	Learning rate 0.001, beta_1=0.9, beta_2=0.999, and epsilon=1e-07
7.	Loss Function	categorical_crossentry	Loss functions for multi-class classification.
8.	Metrics	accuracy	Metrics that the model will evaluate during training and validation.

- The XGB model design used is presented in Table 3.

Table 3. XGB Model Design.

No	Setting/Parameter	Value	Note
1.	n_estimators	150	The number of trees constructed.
	learning_rate	0.01	Step size learning to update model weights.
2.	max_depth	6	Maximum depth of each tree.
3.	random_state	42	Seeds for reproduction
4.	eval_metric	["error", "logloss"]	Metrics for evaluating model performance during training
5.	eval_set	(X_val, y_val)	Dataset used for evaluation of model performance during training.
6.	verbose	True	Determines whether evaluation metric output is printed during training.
7.	n_estimators	150	The number of trees constructed.

4. Model Training: The model is trained using the fit method on training data that has been scaled and balanced. Cross-validation was also done with a 10% validation subset of the training data set. The result will be the predicted probability for each model, namely BiGRU, BiLSTM, and XGB.
5. Ensemble predictions are carried out using the following steps:
   - Prediction Probability Extraction: Prediction probabilities from BiGRU, BiLSTM models, and positive class probabilities from XGBoost are extracted.
   - Meta-learner training is carried out based on feature stacks from previously made model predictions so as to produce continuous predictions from XGBRegressor. XGBRegressor parameters are presented in Table 4.

Table 4. XGBRegressor Model Design.

No	Setting/Parameter	Value	Note
1.	n_estimators	100	The number of trees constructed.
	learning_rate	0.1	Step size learning to update model weights.
2.	max_depth	3	Maximum depth of each tree.
3.	objective	reg:squarederror	The objective function used for training.
4.	booster	gbtree	Gradient boosting based trees
5.	n_estimators	100	The number of trees constructed.

- The results of the final predictions are rounded to the nearest integer, which may indicate the prediction class in the case of classification. Then, the output is converted into classification results.

6. After training the ensemble model, the testing data is tested with the model, and then the accuracy, precision, recall, f1, and specificity are calculated.

4. Results and Discussion

In this section, the proposed method is tested with two diabetes datasets, namely ISD [7] and PID [9], [10]. These two datasets were chosen because they are the two most popular datasets; however, as previously discussed, this research focuses more on the PID dataset because it is relatively more challenging than PID. More detailed features of the PID and ISD datasets are presented, respectively, in Tables 5 and 6.

Table 5. PID Dataset Details

No	Features	Note
1.	Pregnancies	Number of pregnancies
2.	Glucose	2-hour plasma glucose concentration in the oral glucose tolerance test
3.	BloodPressure	Diastolic blood pressure (mm Hg)
4.	SkinThickness	Triceps skinfold thickness (mm)
5.	Insulin	2-hour serum insulin (mu U/ml)
6.	BMI	Body mass index (weight in kg/(height in m)2)
7.	DiabetesPedigree-Function	Function of diabetes pedigree
8.	Age	Age (years)
9.	Outcome	Classification results (0 or 1, where 1 indicates diabetes and 0 does not)

Table 6. ISD Dataset Details

No	Features	Note
1.	ID	Unique identification for each record.
2.	No_Patien	The patient number may be another form of identification.
3.	Gender	Patient gender (F for female, M for male).
4.	Age	Patient age.
5.	Urea	Urea level in the blood.
6.	Cr (Creatinine)	Creatinine level in the blood.
7.	HbA1c	Hemoglobin A1c (long-term blood sugar control indicator).
8.	Chol (Cholesterol)	Total cholesterol level.
9.	TG (Triglycerides)	Triglyceride level.
10.	HDL(High-Density Lipoprotein)	Good cholesterol.
11.	LDL (Low-Density Lipoprotein)	Bad cholesterol.
12.	VLDL (Very LDL)	Bad cholesterol.
13.	BMI (Body Mass Index)	Body mass index.
14.	Class	'N' for normal, 'Y' for diabetes, 'P' for pre-diabetes

The PID and ISD datasets have a different focus and data composition, influencing their approach to research and development of predictive models for type 2 diabetes. The PID dataset is focused on Pima women over 21 years of age, primarily due to the high prevalence of type 2 diabetes in this group and its association with risk factors such as gestational diabetes[38]. The data included are the number of pregnancies, plasma glucose, blood pressure, and other variables useful in identifying the risk of type 2 diabetes. PID can be more difficult to predict with its homogeneity and limited variables, even though this dataset is clean with no missing values or data. Duplicates, with a distribution of 500 records for no diabetes and 268 for diabetes[39].

In contrast, the ISD dataset captures a more diverse population from Iraq, with broader data, including lipid profiles and HbA1c. This diversity allows the creation of more robust predictive models, but like PID, ISD also has no missing values or duplicates, showing good data cleanliness. The class distribution on the ISD was 103 entries for class 'N', 844 for 'Y', and 53 for 'P', indicating an unequal distribution similar to PID. To overcome this imbalance, both datasets apply the SMOTE-Tomek technique to 80% of the training dataset to improve the quality of machine learning. The results of this resampling, which aims to provide a more balanced class distribution, are presented in Figure 2.

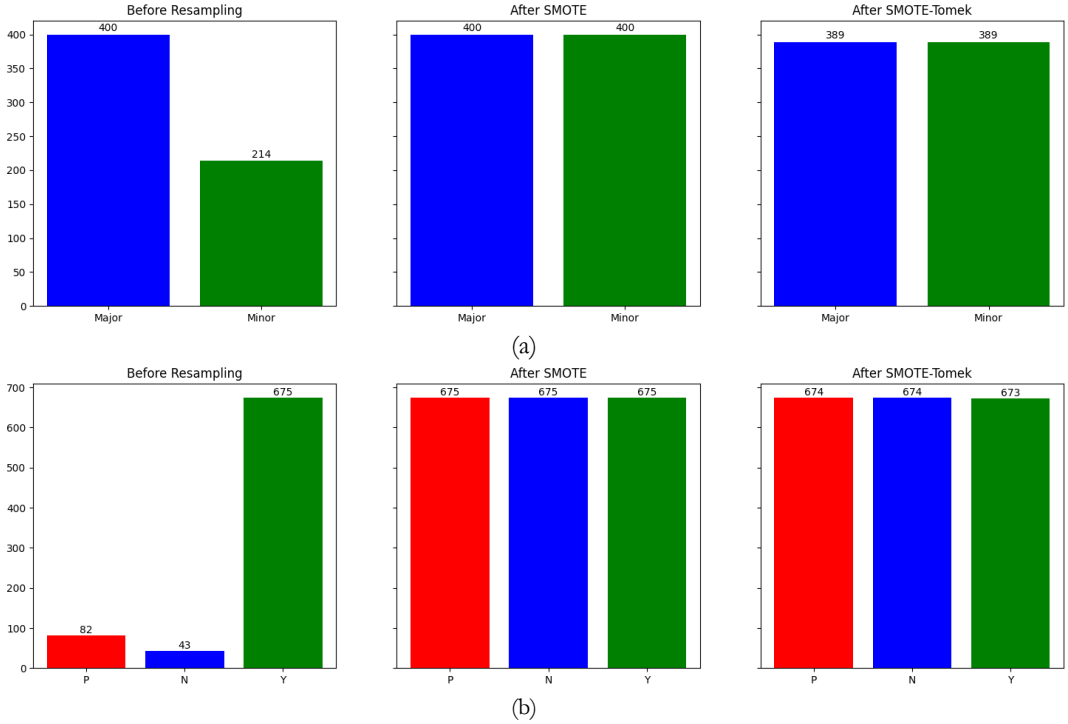

Figure 1. Before after SMOTE-Tomek (**a**)PID Dataset; (**b**) ISD Dataset.

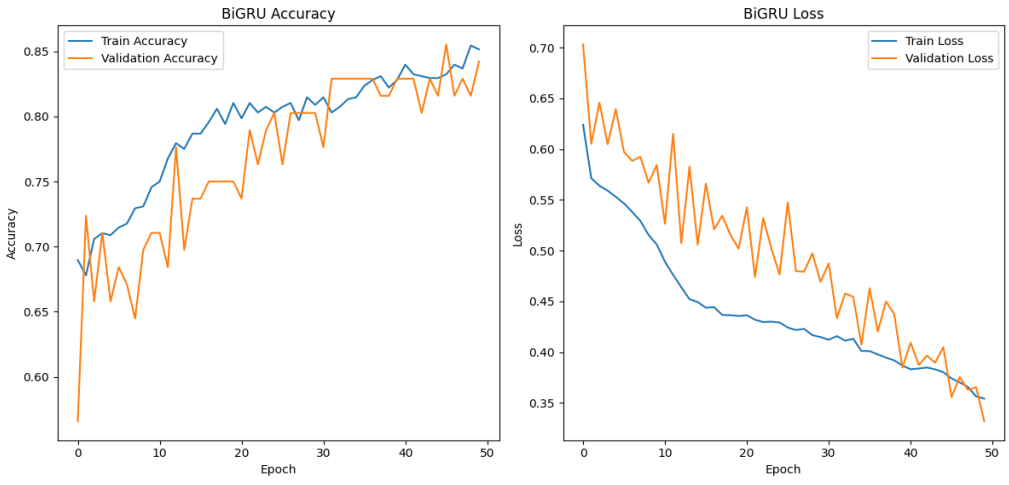

Figure 2. BiGRU Model Prediction for PID Dataset

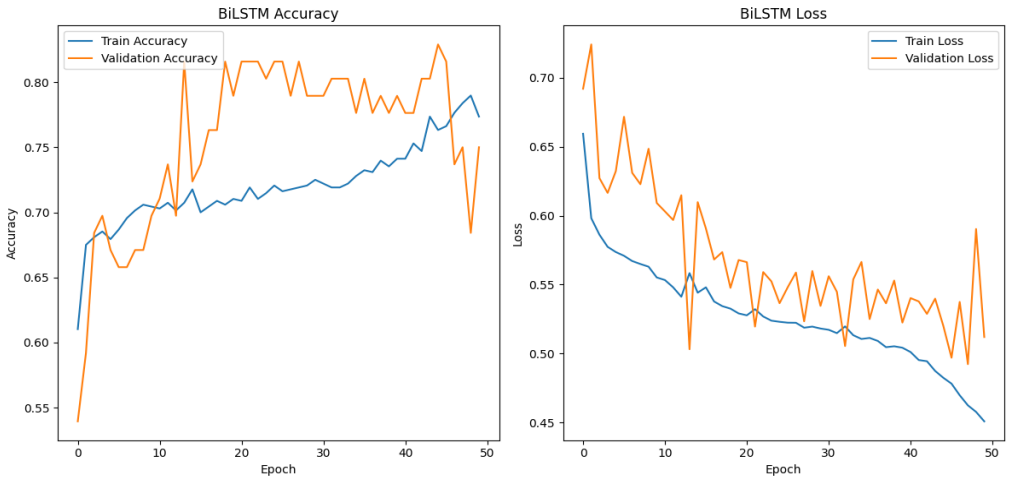

Figure 3. BiLSTM Model Prediction for PID Dataset

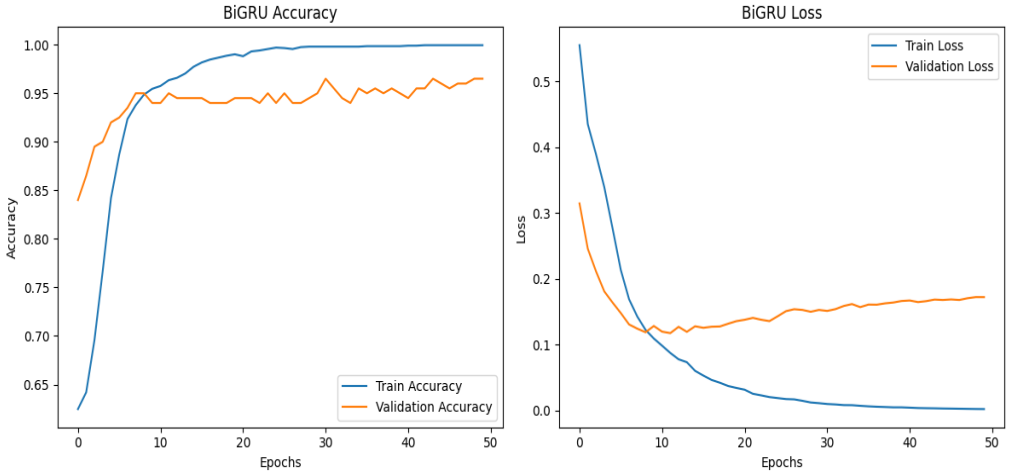

Figure 4. BiGRU Model Prediction for ISD Dataset

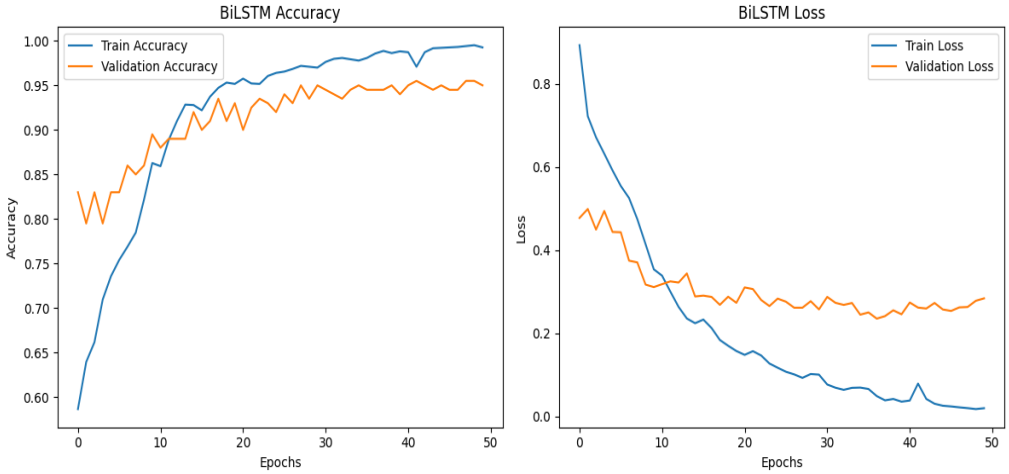

Figure 5. BiLSTM Model Prediction for ISD Dataset

After carrying out resampling, normalization was carried out using a standard scaler. Then the training and validation process was carried out on the three base learners, namely BiGRU, BiLSTM, and XGB. Accuracy and loss plots for each model are presented in Figure 3 for BiGRU on the PID dataset and Figure 5 for the ISD dataset. Meanwhile, the BiLSTM method plots in Figure 4 for the PID dataset and Figure 6 for the ISD dataset. In more detail,

the results of all base learners are presented in Table 5. This table shows that the performance of XGB is the best on both datasets, followed by BiGRU and BiLSTM. The performance of BiGRU and BiLSTM may not be special. This is because the design of BiLSTM and BiGRUM is relatively simple, with only three layers. The purpose of this layer's simplicity is to reduce computational complexity, considering that the proposed method uses three base learners, two of whom are deep learning methods. Because the stacking ensemble can combine the performance of all three base predictors, we reduce the complexity of the BiGRU and BiLSTM models. So even though the prediction results of each model, especially BiGRU and BiLSTM, appear relatively weak (see Table 5), after being combined with an ensemble, the method can recognize the diabetes dataset powerfully and accurately (see Figures 7 and 8).

Table 5. Base Learner Results Details

Dataset	Method	Acc_train	Acc_val	Loss_train	Loss_val
PID	BiGRU	0.8515	0.8421	0.3543	0.3319
	BiLSTM	0.7735	0.7500	0.4508	0.5120
	XGB	0.9623	-	-	-
ISD	BiGRU	0.9995	0.9650	0.0022	0.1723
	BiLSTM	0.9926	0.9500	0.0192	0.2835
	XBG	0.9912	-	-	-

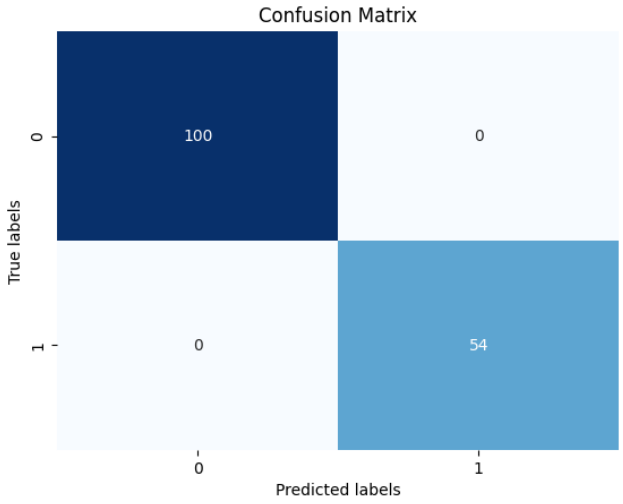

Figure 6. Final prediction for PID dataset using the proposed method

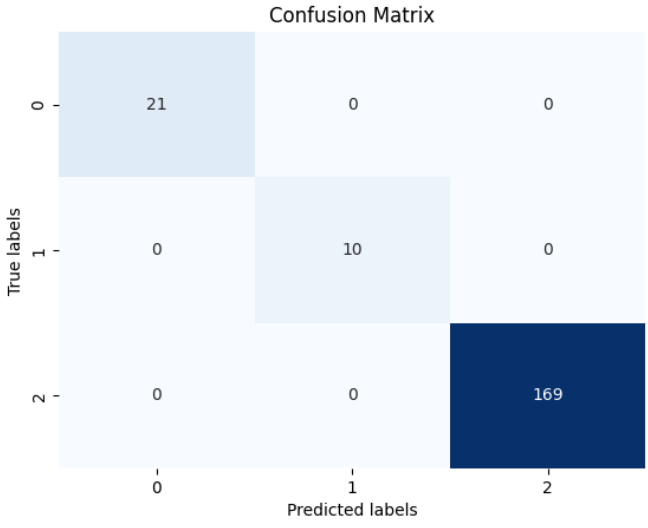

Figure 7. Final prediction for ISD dataset using the proposed method

From the results presented in Table 5, it appears that XGB performs better than BiGRU and BiLSTM on both datasets. BiGRU and BiLSTM show lower performance, possibly because these deep learning models require large datasets to perform optimally. In this context, simpler designs aim to reduce computational overhead, considering the use of multiple base learners in one ensemble. However, the results of the three base learners above can still be maximized with a stacking-boosting ensemble, where it appears that all testing results on the two datasets produce an accuracy of 1.0, and this means that the same values are obtained for precision, recall, specificity, and f1. This is because using XGB as a meta-learner in the stacking ensemble allows more effective integration of predictions from the base learner. XGB performs regularization and handles overfitting efficiently, ensuring the resulting predictions are accurate and generalizable. The proposed method can also reduce bias and variance, producing a more stable and robust model for new data and hidden variables in the training dataset.

5. Comparison

In this section, we explore the high performance of our proposed method across two different datasets, demonstrating its robustness and precision. We also compared several previous methods that used the same dataset, as shown in Table 6.

Table 6. Comparison testing results with prior art

Dataset	Method	Accuracy	Recall	Precision	F1	Specificity
PID	Ref [2]	0.7710	0.70	0.68	0.69	-
	Ref [12]	0.78	0.85	0.81	0.83	-
	Ref [4]	0.7957	0.8133	0.8940	0.8517	0.7500
	Ref [13]	0.871	0.857	0.806	0.830	-
	Ref [17]	0.9342	0.9767	0.9545	0.9655	0.9394
	Ref [23]	0.935	0.85	-	-	0.98
	Ref [1](2GDNN + O2GDNN)	0.97248	0.97245	0.97342	0.97255	-
	Ref [1] (RF + ORF)	0.97931	0.97931	0.98119	0.97954	-
	Ref [37]	0.9807	0.9846	0.9522	0.9681	-
	Ours Method	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
ISD	Ref [1](2GDNN + O2GDNN)	0.97333	0.97333	0.97281	0.97265	-
	Ref [12]	0.99	1.00	0.94	0.97	-
	Ref [1] (RF + ORF)	**1.0000**	**1.0000**	**1.0000**	**1.0000**	-
	Ours Method	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

The ISD dataset has proven easier to recognize, as evidenced by the performance of the base learner in refs [1] and [12]. Although research [1] also succeeded in getting perfect performance on the ISD dataset, on the PID dataset, the accuracy, recall, and f1 were around 0.97, and the precision was around 0.98. However, if we focus on recall performance and accuracy, the method [37] is superior to reference [1]. In medical practice, the choice of model evaluation metrics is strongly influenced by the consequences of diagnostic errors. Recall (sensitivity) and specificity are two very critical metrics because they are directly related to patient clinical outcomes. High recall is essential in a medical context because it ensures that the model identifies almost all positive cases, such as serious illnesses. Failure to detect positive cases may result in not implementing necessary treatment, worsening the patient's condition and increasing the risk of serious complications. Therefore, high recall helps start treatment as quickly as possible, vital for diseases with serious health implications, such as cancer or heart disease. On the other hand, high specificity reduces the possibility of wrong diagnosis in healthy individuals. Low specificity leads to many "false positives," in which individuals who do not have the disease are considered patients, resulting in unnecessary anxiety, further medical testing, and potentially risky interventions. High specificity is important to avoid these costs and risks, ensuring that only those who genuinely need treatment receive further intervention[27], [40].

Finding a balance between recall and specificity is important because placing too much emphasis on one can come at the expense of the other. For example, increasing recall may

decrease specificity, which may be undesirable in certain medical conditions. Therefore, these two metrics are often weighed in medical settings based on clinical priorities and the consequences of diagnostic errors. Additionally, accuracy can provide a general idea of model reliability but may be less informative in imbalanced datasets, where most classes can distort the perception of model performance. Precision and F1 scores are also important, as precision indicates the accuracy of positive predictions, and the F1 score balances precision and recall. In practice, F1 scores are often used to assess model performance on imbalanced datasets, providing a more holistic insight into a model's effectiveness in identifying positive cases without overpredicting false positives.

6. Conclusions

This research succeeded in developing a robust method for diabetes classification by combining techniques from deep learning and ensemble learning, especially using the SMOTE-Tomek method for data balancing and XGBoost as a meta-learner in the stacking framework. Using BiLSTM, BiGRU, and XGBoost as base learners shows that integrating these approaches can increase accuracy, precision, recall, and model specificity. The final results confirm that ensemble techniques with a rigorous meta-learner can minimize the individual weaknesses of each model and improve generalization on complex and imbalanced medical data. This indicates the importance of a hybrid approach in developing medical diagnostic tools, especially in the face of the wide and inconsistent data variance often found in health datasets.

Author Contributions: Conceptualization: D.R.I.M.S.; Methodology: D.R.I.M.S.; Software: K.N.; Validation: All; Formal analysis: A.R.M and S.W.I; Investigation: K.N., A.R.M and S.W.I; Resources: A.R.M and S.W.I.; Data curation: K.N.; Writing—original draft preparation: D.R.I.M.S.; Writing—review and editing: A.A.O.; Visualization: A.R.M.; Supervision: D.R.I.M.S.; Project administration: K.N.; Funding acquisition: All.

Data Availability Statement: Our code URL https://github.com/MosesdeRosal/Robust-Diabetes-Recognition-using-Stacking-Ensemble-XGBoost-Meta-Learner.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] C. C. Olisah, L. Smith, and M. Smith, "Diabetes mellitus prediction and diagnosis from a data preprocessing and machine learning perspective," *Comput. Methods Programs Biomed.*, vol. 220, p. 106773, 2022, doi: 10.1016/j.cmpb.2022.106773.

[2] M. S. Reza, R. Amin, R. Yasmin, W. Kulsum, and S. Ruhi, "Improving diabetes disease patients classification using stacking ensemble method with PIMA and local healthcare data," *Heliyon*, vol. 10, no. 2, p. e24536, 2024, doi: 10.1016/j.heliyon.2024.e24536.

[3] F. Mustofa, A. N. Safriandono, A. R. Muslikh, and D. R. I. M. Setiadi, "Dataset and Feature Analysis for Diabetes Mellitus Classification using Random Forest," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 41–48, Jan. 2023, doi: 10.33633/jcta.v1i1.9190.

[4] V. Chang, J. Bailey, Q. A. Xu, and Z. Sun, "Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms," *Neural Comput. Appl.*, vol. 35, no. 22, pp. 16157–16173, Aug. 2023, doi: 10.1007/s00521-022-07049-z.

[5] M. Zhao, J. Wan, W. Qin, X. Huang, G. Chen, and X. Zhao, "A machine learning-based diagnosis modelling of type 2 diabetes mellitus with environmental metal exposure," *Comput. Methods Programs Biomed.*, vol. 235, p. 107537, Jun. 2023, doi: 10.1016/j.cmpb.2023.107537.

[6] L. Wang, Z. Pan, W. Liu, J. Wang, L. Ji, and D. Shi, "A dual-attention based coupling network for diabetes classification with heterogeneous data," *J. Biomed. Inform.*, vol. 139, no. July 2022, p. 104300, 2023, doi: 10.1016/j.jbi.2023.104300.

[7] A. Rashid, "Diabetes Dataset." Mendeley Data, 2020. doi: 10.17632/wj9rwkp9c2.1.

[8] "Abelvikas Diabetes Dataset." https://data.world/abelvikas/diabetes-type-dataset (accessed Dec. 12, 2020).

[9] UCI Machine Learning, "Pima Indians Diabetes Database," *Kaggle.com*. https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[10] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," *Proc. - Annu. Symp. Comput. Appl. Med. Care*, pp. 261–265, 1988.

[11] O. Adigun, F. Okikiola, N. Yekini, and R. Babatunde, "Classification of Diabetes Types using Machine Learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 9, pp. 152–161, 2022, doi: 10.14569/IJACSA.2022.0130918.

[12] X. Li, M. Curiger, R. Dornberger, and T. Hanne, "Optimized computational diabetes prediction with feature selection algorithms," *ACM Int. Conf. Proceeding Ser.*, no. Ml, pp. 36–43, 2023, doi: 10.1145/3596947.3596948.

[13] Q. Wang, W. Cao, J. Guo, J. Ren, Y. Cheng, and D. N. Davis, "DMP_MI: An Effective Diabetes Mellitus Classification Algorithm on Imbalanced Data With Missing Values," *IEEE Access*, vol. 7, pp. 102232–102238, 2019, doi: 10.1109/ACCESS.2019.2929866.
[14] N. Pradhan, G. Rani, V. S. Dhaka, and R. C. Poonia, "Diabetes prediction using artificial neural network," in *Deep Learning Techniques for Biomedical and Health Informatics*, Elsevier, 2020, pp. 327–339. doi: 10.1016/B978-0-12-819061-6.00014-8.
[15] Asniar, N. U. Maulidevi, and K. Surendro, "SMOTE-LOF for noise identification in imbalanced data classification," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3413–3423, Jun. 2022, doi: 10.1016/j.jksuci.2021.01.014.
[16] I. Tasin, T. U. Nabil, S. Islam, and R. Khan, "Diabetes prediction using machine learning and explainable AI techniques," *Healthc. Technol. Lett.*, vol. 10, no. 1–2, pp. 1–10, Feb. 2023, doi: 10.1049/htl2.12039.
[17] E. Pekel Özmen and T. Özcan, "Diagnosis of diabetes mellitus using artificial neural network and classification and regression tree optimized with genetic algorithm," *J. Forecast.*, vol. 39, no. 4, pp. 661–670, Jul. 2020, doi: 10.1002/for.2652.
[18] M. A. Araaf, K. Nugroho, and D. R. I. M. Setiadi, "Comprehensive Analysis and Classification of Skin Diseases based on Image Texture Features using K-Nearest Neighbors Algorithm," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 31–40, Sep. 2023, doi: 10.33633/jcta.v1i1.9185.
[19] S. Ali, A. Hashmi, A. Hamza, U. Hayat, and H. Younis, "Dynamic and Static Handwriting Assessment in Parkinson's Disease: A Synergistic Approach with C-Bi-GRU and VGG19," *J. Comput. Theor. Appl.*, vol. 1, no. 2, pp. 151–162, Dec. 2023, doi: 10.33633/jcta.v1i2.9469.
[20] A. Yazdizadeh, Z. Patterson, and B. Farooq, "Ensemble Convolutional Neural Networks for Mode Inference in Smartphone Travel Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2232–2239, 2020, doi: 10.1109/TITS.2019.2918923.
[21] K. Jhang, "Voting and ensemble schemes based on CNN models for photo-based gender prediction," *J. Inf. Process. Syst.*, vol. 16, no. 4, pp. 809–819, 2020, doi: 10.3745/JIPS.02.0137.
[22] A. Manconi, G. Armano, M. Gnocchi, and L. Milanesi, "A Soft-Voting Ensemble Classifier for Detecting Patients Affected by COVID-19," *Appl. Sci.*, vol. 12, no. 15, 2022, doi: 10.3390/app12157554.
[23] H. Qi, X. Song, S. Liu, Y. Zhang, and K. K. L. Wong, "KFPredict: An ensemble learning prediction framework for diabetes based on fusion of key features," *Comput. Methods Programs Biomed.*, vol. 231, p. 107378, Apr. 2023, doi: 10.1016/j.cmpb.2023.107378.
[24] O. Jaiyeoba, E. Ogbuju, O. T. Yomi, and F. Oladipo, "Development of a Model to Classify Skin Diseases using Stacking Ensemble Machine Learning Techniques," *J. Comput. Theor. Appl.*, vol. 2, no. 1, pp. 22–38, May 2024, doi: 10.62411/jcta.10488.
[25] T. R. Noviandy, K. Nisa, G. M. Idroes, I. Hardi, and N. R. Sasmita, "Classifying Beta-Secretase 1 Inhibitor Activity for Alzheimer's Drug Discovery with LightGBM," *J. Comput. Theor. Appl.*, vol. 1, no. 4, pp. 358–367, Mar. 2024, doi: 10.62411/jcta.10129.
[26] F. Omoruwou, A. A. Ojugo, and S. E. Ilodigwe, "Strategic Feature Selection for Enhanced Scorch Prediction in Flexible Polyurethane Form Manufacturing," *J. Comput. Theor. Appl.*, vol. 1, no. 3, pp. 346–357, Feb. 2024, doi: 10.62411/jcta.9539.
[27] F. S. Gomiasti, W. Warto, E. Kartikadarma, J. Gondohanindijo, and D. R. I. M. Setiadi, "Enhancing Lung Cancer Classification Effectiveness Through Hyperparameter-Tuned Support Vector Machine," *J. Comput. Theor. Appl.*, vol. 1, no. 4, pp. 396–406, Mar. 2024, doi: 10.62411/jcta.10106.
[28] Z. Yishun, W. Guoyue, L. Yi, M. Yige, and W. Jiangwei, "Classification of Distribution Network Planning Documents Based on LSTM Neural Network," *Procedia Comput. Sci.*, vol. 228, pp. 914–919, 2023, doi: 10.1016/j.procs.2023.11.120.
[29] S. Boda, M. Mahadevappa, and P. Kumar Dutta, "An automated patient-specific ECG beat classification using LSTM-based recurrent neural networks," *Biomed. Signal Process. Control*, vol. 84, no. February 2022, p. 104756, Jul. 2023, doi: 10.1016/j.bspc.2023.104756.
[30] N. N. Wijaya, D. R. I. M. Setiadi, and A. R. Muslikh, "Music-Genre Classification using Bidirectional Long Short-Term Memory and Mel-Frequency Cepstral Coefficients," *J. Comput. Theor. Appl.*, vol. 1, no. 3, pp. 243–256, Jan. 2024, doi: 10.62411/jcta.9655.
[31] J. Bi, Z. Guan, H. Yuan, and J. Zhang, "Improved network intrusion classification with attention-assisted bidirectional LSTM and optimized sparse contractive autoencoders," *Expert Syst. Appl.*, vol. 244, no. December 2023, p. 122966, Jun. 2024, doi: 10.1016/j.eswa.2023.122966.
[32] Y. Lu, X. Wu, P. Liu, H. Li, and W. Liu, "Rice disease identification method based on improved CNN-BiGRU," *Artif. Intell. Agric.*, vol. 9, pp. 100–109, Sep. 2023, doi: 10.1016/j.aiia.2023.08.005.
[33] M. Diaz, M. Moetesum, I. Siddiqi, and G. Vessio, "Sequence-based dynamic handwriting analysis for Parkinson's disease detection with one-dimensional convolutions and BiGRUs," *Expert Syst. Appl.*, vol. 168, no. August 2020, p. 114405, Apr. 2021, doi: 10.1016/j.eswa.2020.114405.
[34] M. R. Abbasniya, S. A. Sheikholeslamzadeh, H. Nasiri, and S. Emami, "Classification of Breast Tumors Based on Histopathology Images Using Deep Features and Ensemble of Gradient Boosting Methods," *Comput. Electr. Eng.*, vol. 103, no. 1, p. 108382, Jan. 2022, doi: 10.1016/j.compeleceng.2022.108382.
[35] J. A. ALzubi, B. Bharathikannan, S. Tanwar, R. Manikandan, A. Khanna, and C. Thaventhiran, "Boosted neural network ensemble classification for lung cancer disease diagnosis," *Appl. Soft Comput.*, vol. 80, pp. 579–591, Jul. 2019, doi: 10.1016/j.asoc.2019.04.031.
[36] E. B. Wijayanti, D. R. I. M. Setiadi, and B. H. Setyoko, "Dataset Analysis and Feature Characteristics to Predict Rice Production based on eXtreme Gradient Boosting," *J. Comput. Theor. Appl.*, vol. 1, no. 3, pp. 299–310, Feb. 2024, doi: 10.62411/jcta.10057.
[37] H. Naz and S. Ahuja, "Deep learning approach for diabetes prediction using PIMA Indian dataset," *J. Diabetes Metab. Disord.*, vol. 19, no. 1, pp. 391–403, 2020, doi: 10.1007/s40200-020-00520-5.
[38] A. Tuppad and S. Devi Patil, "An efficient classification framework for Type 2 Diabetes incorporating feature interactions," *Expert Syst. Appl.*, vol. 239, no. April 2023, p. 122138, 2024, doi: 10.1016/j.eswa.2023.122138.
[39] L. P. Joseph, E. A. Joseph, and R. Prasad, "Explainable diabetes classification using hybrid Bayesian-optimized TabNet architecture," *Comput. Biol. Med.*, vol. 151, no. PA, p. 106178, 2022, doi: 10.1016/j.compbiomed.2022.106178.
[40] Muljono, S. A. Wulandari, H. Al Azies, M. Naufal, W. A. Prasetyanto, and F. A. Zahra, "Breaking Boundaries in Diagnosis: Non-Invasive Anemia Detection Empowered by AI," *IEEE Access*, vol. 12, pp. 9292–9307, Jan. 2024, doi: 10.1109/ACCESS.2024.3353788.