JURNAL LOCUS: Penelitian & Pengabdian
Volume 4 No. 7 July 2025
E-ISSN 2829-7334| P-ISSN 2829-5439
Hompage: https://locus.rivierapublishing.id/index.php/jl

Diabetes Mellitus Prediction Using Decision Tree and Deep Neural Network a Case Study at RSUP Kariadi

Niken Puspitasari1, Aji Supriyanto2
Universitas STIKUBANK, Semarang, Indonesia
Email: nikenpuspitasari0008@mhs.unisbank.ac.id, 2ajisup@edu.unisbank.ac.id

Abstract:
The demands of workload, combined with an unhealthy lifestyle and lack of regular exercise, can lead to diabetes mellitus (*diabetes*), a disease caused by dysfunction of the *pancreas* and *insulin* regulation. Diabetes is the seventh leading cause of death worldwide and ranks as the third-highest cause of mortality in Indonesia. This study aims to identify the most effective predictive model for diabetes diagnosis using health technology. Machine learning approaches, including ensemble methods and deep learning, were applied to data collected from patients at Dr. Kariadi Central General Hospital, where there is a significant imbalance between positive and negative *diabetes* cases. The dataset was processed using SMOTE (*Synthetic Minority Over-sampling Technique*), and models were optimized through hyperparameter tuning to maximize accuracy, precision, recall, and F1-score. The proportion of non-diabetic patients is 91.49%, which serves as the baseline for developing the automated detection system. In this context, accuracy alone is insufficient; precision and recall are more critical for reliable diabetes prediction. Low recall means many diabetic patients go undetected (high *False Negative* rate), risking delayed treatment. Low precision indicates many healthy individuals are misdiagnosed (high *False Positive* rate), causing unnecessary anxiety and intervention. The Decision Tree with Ensemble Extra Trees model achieved the highest recall at 99.65%, while the Deep Neural Network model yielded the highest precision at 99.15%. The best balance, as indicated by the F1-score, was achieved by the Decision Tree with Ensemble Extra Trees model, scoring 99.39%. These findings highlight the importance of advanced health technology in improving early detection and management of diabetes.

Keywords: Diabetes Mellitus, Deep Learning, Deep Neural Network, Machine Learning, Machine Learning with Ensemble..

Corresponding: Nama author
E-mail: Email author

INTRODUCTION

The demands of workload combined with an unhealthy lifestyle, without being balanced by regular exercise, can lead to diabetes mellitus (diabetes), a disease caused by problems with the pancreas and insulin (Purbolaksono, M. D., Tantowi, I., Hidayat, A., & Adiwijaya, 2021). Diabetes is a major issue that threatens public health and economic stability in both developing and developed countries (Fauzi, A., & Isnawati, 2023). 55.29 million people worldwide suffer from diabetes (Taylor, R. B., & Gatta, F., n.d.). Diabetes is a long-term disease, and many people who suffer from it do not realize how serious their condition is early on (García-Ordás et al., 2021). Machine learning and deep learning predictive models for type 2 diabetes (Fregoso-Aparicio et al., 2021).

Diabetes prediction using Deep Learning can achieve a high level of accuracy (Kumar P. B. M. et al., 2020; Rahman et al., 2020). Feature selection approaches to improve the accuracy and reliability of diabetes identifications (Wee et al., 2024). LSTM model achieves its highest accuracy in diabetes detection (Mousa et al., 2023). Decision tree models are well-suited for diabetes detection (Sharma, T., & Shah, M., 2021; Sunge et al., 2019). Using SMOTE-based deep LSTM method for diabetes can predict highest prediction accuracy (Alex et al., 2022). The diabetic data set was subjected to the proposed DBN approach, which had an accuracy of 81.25%. Compared to other machine learning techniques, the suggested method produces results with higher accuracy (Shahin et al., 2023). Cross-validation techniques that can minimize overfitting (Prasetyo, A. B., & Laksana, T. G., 2022). The use of SMOTE in Deep Learning models can improve accuracy (Zamachsari & Puspitasari, 2021).

Early detection in identifying diabetes is one of the main objectives of deep learning-based diabetes diagnosis (Zahir & Saputra, 2024). Several classification machine learning models have also been compared, such as Decision Trees, Random Forest, Extra Trees Classifier, MLP, and SVM, as well as ensemble methods, in terms of the accuracy achieved (Abhishek, L., 2020). The random forest classifier model produced an accuracy of 82%. The resampling techniques used Random Oversampling (ROS), Random Under sampling (RUS), Synthetic Minority Over-sampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), Borderline SMOTE, SMOTE + Edited Nearest Neighbors (ENN), SMOTE + Tomek Links (Brownlee, n.d.).

Overfitting and long training time are two fundamental challenges in multilayered neural network learning and deep learning in particular. Dropout and batch normalization are two well-recognized approaches to tackle these challenges (Garbin et al., 2020). Oversampling and Under sampling to Address the Problem of Imbalanced Dataset (Indrawati, A., 2021). Dropout and its variant drop connect could improve performance of shallow multi-layer perceptron neural networks (Piotrowski et al., 2020).

Diabetes mellitus is a chronic disease that poses a serious threat to global health, including in Indonesia where it ranks as the third leading cause of death. The combination of work demands, unhealthy lifestyles, and lack of physical activity has contributed to the increasing prevalence of diabetes. Early detection is crucial to prevent more severe complications. In recent years, advancements in artificial intelligence technology, particularly machine learning and deep learning, have opened new opportunities for more accurate diabetes prediction. However, the main challenges in developing these predictive models include data imbalance between diabetic and non-diabetic patients, as well as the need to minimize diagnostic errors, especially false negatives which can have fatal consequences.

Several previous studies have attempted to address these issues. García-Ordás et al. (2021) proposed using deep learning with oversampling techniques and feature augmentation, but their approach lacked in-depth consideration of data imbalance aspects and the clinical impact of false negatives. On the other hand, Fregoso-Aparicio et al. (2021) in their systematic review highlighted the lack of focus on clinical metrics such as recall, which is particularly important in diabetes diagnosis. They also identified gaps in hyperparameter optimization, especially for datasets from specific populations like Southeast Asia.

The aim of this study is to obtain the best model for implementation in a diabetes mellitus prediction system. The goal is to detect diabetes mellitus early, allowing patients to obtain preliminary information about their condition before undergoing clinical testing. Machine learning with ensemble methods and deep learning are used for diabetes mellitus prediction. Both models will be configured and optimized using SMOTE to achieve the best accuracy score, precision score, and recall score.

RESEARCH METHODS

Diabetes mellitus is a chronic disease characterized by elevated blood sugar levels due to impaired *insulin* utilization. The research process for predicting diabetes involves several essential steps: data understanding, pre-processing, model application, evaluation, and deployment, primarily using Python. With increasing diabetes cases linked to unhealthy lifestyles, early detection through data analysis is crucial, as highlighted by the Indonesian Ministry of Health, which reports a 6.7% mortality rate.

The dataset, obtained from patients at Dr. Kariadi Central General Hospital, comprises 1,846 medical records and ten variables relevant to diabetes prediction, such as age, BMI, blood pressure, and family history. Data pre-processing is vital for improving model accuracy by resolving inconsistencies, addressing missing values, and converting categorical variables into numerical formats. Feature selection and resampling techniques, like SMOTE (*Synthetic Minority Over-sampling Technique*), are employed to balance the dataset and enhance predictive performance.

For model development, machine learning algorithms such as Decision Trees and ensemble methods like Random Forest are utilized, alongside deep learning approaches that leverage Keras Tuner for hyperparameter optimization. These models aim to accurately predict diabetes outcomes by learning complex patterns within the medical data. Model effectiveness is assessed using evaluation metrics including accuracy, precision, recall, and F1-score, ensuring the predictions are both reliable and actionable for clinical decision-making. This comprehensive approach underscores the importance of health technology in facilitating early intervention and improved management of diabetes.

RESULTS AND DISCUSSIONS

3.1. Data Understanding

3.1.1. Analysis of Diabetes Data Distribution

The variables, data types, number of data points, and data clusters are presented in Table 2.

Table 2. Diabetes Data Structure

No.	Variable	Data Type	Count	Unique	Missing	Pc_missing
1	Sex	object	1,846	2	0	0.00
2	Age	int64	1,846	6	0	0.00
3	BMI	int64	1,846	6	0	0.00
4	BP	int64	1,846	5	0	0.00
5	BPM	object	1,846	2	0	0.00
6	RR	object	1,846	2	0	0.00
7	Temp	int64	1,846	4	0	0.00
8	DMFam	object	1,846	3	3	0.16
9	HTA	object	1,846	3	8	0.43
10	Diabetes	object	1,846	2	0	0.00

Table 2 explains that there are four variables with two-class values. The Age and BMI variables have six classes, the BP variable has five classes, the Temp variable has four classes, and the DMFam and HTA variables have three classes. However, there are missing values, with DMFam having 3 missing data points and HTA having 8 missing data points.

The graphical representation of diabetes data variables, including Age, BMI, BP, and Temp, in relation to Diabetes values, is presented in figure 4.

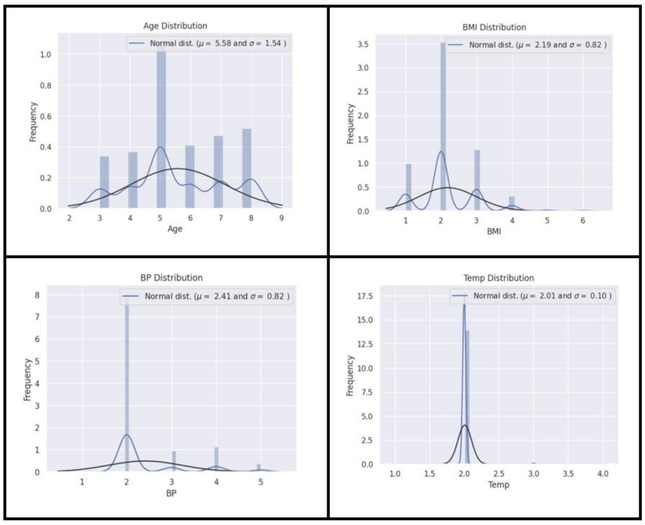

Figure 4. Diabetes Data Variable Graph

3.1.2. Diabetes Disease Analysis

The diabetes dataset, consisting of 1,846 patients, is imbalanced between those with and without diabetes. The comparison of diabetes data is presented in Figure 5.

Figure 5. Diabetes Data Comparison

The number of data points with the non-diabetes target class is 1,689, while the number of diabetes cases is 157. In percentage terms, 91.49% of the data represents non-diabetic patients, whereas 8.51% represents diabetic patients. From this data, we can derive an important insight: if we simply predict that all patients do not have diabetes, the accuracy would be 91.49%. This 91.49% accuracy serves as a baseline metric for evaluating the performance of machine learning models.

3.2. Pre Processing

3.2.1. Data Cleansing

To identify null values in the dataset, Python Pandas provides the functions isnull() and isna(). These functions help detect missing values within the dataset. The results of the null value calculations are presented in Table 3.

Table 3. Number of Null Data

No.	Nama Variabel	Jumlah

1	Sex	0
2	Age	0
3	BMI	0
4	BP	0
5	BPM	0
6	RR	0
7	Temp	0
8	DMFam	3
9	HTA	8
10	Diabetes	0

3.2.2. Feature Encoding

This process will use Feature Encoding. The encoding method applied is Label Encoding using the Scikit-Learn library in Python.

table 4. Data After Label Encoding

No	Sex	Age	BMI	BP	BPM	RR	Temp	DMFam	HTA	Diabetes
0	1	5	2	4	1	1	2	0	0	0
1	0	5	2	4	1	1	2	0	0	0
2	0	4	2	3	1	1	2	0	0	1
3	0	5	2	2	1	1	2	0	0	0
4	0	8	2	5	1	1	2	0	0	0
5	0	7	3	3	1	1	2	0	0	0
6	0	7	3	3	1	1	2	0	0	1
7	0	3	1	3	1	1	2	0	0	0
8	0	8	3	2	0	1	2	0	0	1
9	1	4	1	2	1	1	2	0	0	0

3.2.3. Feature Scaling

Table 5. Feature Scaling Result Data

Sex	Age	BMI	BP	BPM	RR	Temp	DMFam	HTA
0.798735	-0.376609	-0.228955	1.938794	0.128529	0.214225	-0.065974	-0.245963	-0.175318
-1.251980	-0.376609	-0.228955	1.938794	0.128529	0.214225	-0.065974	-0.245963	-0.175318
-1.251980	-1.024530	-0.228955	0.721443	0.128529	0.214225	-0.065974	-0.245963	-0.175318
-1.251980	-0.376609	-0.228955	-0.495909	0.128529	0.214225	-0.065974	-0.245963	-0.175318
-1.251980	1.567155	-0.228955	3.156146	0.128529	0.214225	-0.065974	-0.245963	-0.175318
-1.251980	0.919234	0.992578	0.721443	0.128529	0.214225	-0.065974	-0.245963	-0.175318
-1.251980	0.919234	0.992578	0.721443	0.128529	0.214225	-0.065974	-0.245963	-0.175318
-1.251980	-1.672451	-1.450487	0.721443	0.128529	0.214225	-0.065974	-0.245963	-0.175318
-1.251980	1.567155	0.992578	-0.495909	-7.780317	0.214225	-0.065974	-0.245963	-0.175318
0.798735	-1.024530	-1.450487	-0.495909	0.128529	0.214225	-0.065974	-0.245963	-0.175318

If we perform the test in table 5, the maximum standard deviation is 1, and the mean is 0 or very close to 0.

3.2.4. Resampling Dataset

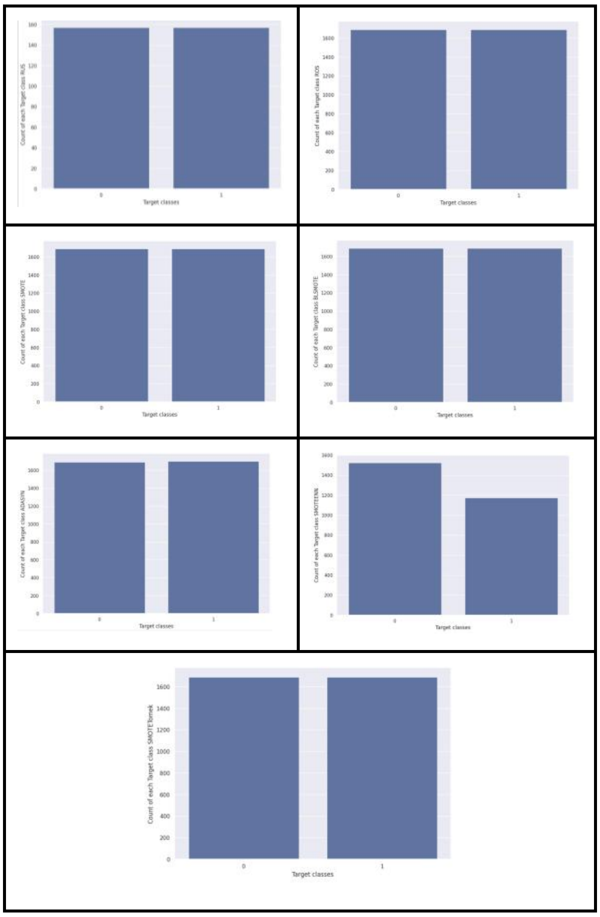

Figure 6. Resampling Dataset Results

Figure 6 shows that resampling using ROS, SMOTE, ADASYN, BLSMOTE, and SMOTETomek results in the same number of samples since they fall under the oversampling category. Meanwhile, resampling using RUS and SMOTEENN produces different sample sizes.

3.2.5. Split Dataset

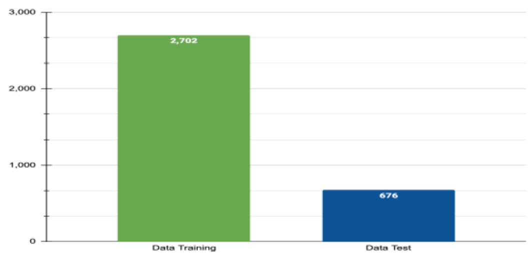

Figure 7. Resampling Dataset Results

Figure 7 illustrates the results of data splitting using one of the resampling mechanisms, Random Oversampling (ROS), which produces 2,702 training data and 676 testing data. The 80% - 20% split into training and testing sets will also be applied to all other resampling methods in the dataset.

3.3. Modeling

3.3.1. Machine Learning

The validation method used is K-fold Cross-Validation, with the number of K set to 10. The random state used is a random state seed. The purpose of the random state seed in machine learning is to ensure reproducibility. The parameter max_features used is 7. Max_features represents the maximum number of features considered to help split branches.

Table 6. Hyperparameter Selection for Decision Tree

Hyperparameter	Description
max_features	7
Random state	seed
Cross-validation	10
Num Tree	50, 100, 200

Table 6 explains the hyperparameter values that will be used to find the best model for each resampled dataset. The resulting model will fall within the range of the specified values.

Table 7. Result Hyperparameter Decision Tree with Ensemble Random Forest

Num Tree	Score	Data Resampling Method						
		RUS (%)	ROS (%)	SMOTE (%)	BLSMOTE (%)	ADASYN (%)	SMOTE ENN (%)	SMOTE Tomek (%)
50	Accuracy	76.84	89.94	91.62	91.72	92.33	99.04	91.52
50	ROC-AUC	77.02	89.96	91.62	91.72	92.33	98.98	91.51
50	Precision	73.08	88.19	91.47	91.16	91.89	99.12	92.29
50	Recall	82.61	92.05	91.65	92.25	92.97	98.60	90.46
50	F1 Score	77.55	90.08	91.56	91.70	92.43	98.86	91.37
100	Accuracy	78.95	89.94	91.52	91.72	92.33	99.19	91.62
100	ROC-AUC	79.19	89.96	91.52	91.72	92.32	99.15	91.62
100	Precision	74.07	88.19	91.45	91.16	91.73	99.13	91.47
100	Recall	86.96	92.05	91.45	92.25	93.16	98.95	91.65
100	F1 Score	80	90.08	91.45	91.70	92.44	99.04	91.56
200	Accuracy	77.89	89.94	91.52	91.72	92.43	99.19	91.62
200	ROC-AUC	78.11	89.96	91.52	91.72	92.42	99.15	91.62
200	Precision	73.58	88.19	91.62	90.99	91.75	99.13	91.47
200	Recall	84.78	92.05	91.25	92.45	93.36	98.95	91.65
200	F1 Score	78.79	90.08	91.43	91.72	92.55	99.04	91.56

Table 7 presents the prediction results for the test data using the Decision Tree model with the Ensemble Random Forest. In the table above, the resampled data using SMOTENN with 200 trees produced the best average values for accuracy, precision, recall, F1 Score, and ROC-AUC.

Table 8. Result Hyperparameter Decision Tree with Ensemble Extra Trees

Num Tree	Score	Data Resampling Method						
		RUS (%)	ROS (%)	SMOTE (%)	BL SMOTE (%)	ADASYN (%)	SMOTE ENN (%)	SMOTE Tomek (%)
50	Accuracy	77.89	89.94	91.72	92.01	92.82	99.48	92.01
50	ROC-AUC	78.11	89.96	91.71	92.02	92.81	99.50	92.01
50	Precision	73.58	88.19	92.15	91.21	91.65	99.13	92.20
50	Recall	84.78	92.05	91.05	92.84	94.34	99.65	91.65
50	F1 Score	78.79	90.08	91.60	92.02	92.97	99.39	91.92
100	Accuracy	77.89	89.94	92.11	92.01	92.92	99.04	92.01
100	ROC-AUC	78.11	89.96	92.11	92.02	92.91	99.00	92.01
100	Precision	73.58	88.19	92.22	91.21	91.67	98.95	92.20
100	Recall	84.78	92.05	91.85	92.84	94.53	98.78	91.65
100	F1 Score	78.79	90.08	92.03	92.02	93.08	98.86	91.92

200	Accuracy	76.84	89.94	91.81	92.01	93.02	99.19	92.01
200	ROC-AUC	77.02	89.96	91.81	92.02	93.01	99.15	92.01
200	Precision	73.08	88.19	92.17	91.21	91.68	99.13	92.20
200	Recall	82.61	92.05	91.25	92.84	94.73	98.95	91.65
200	F1 Score	77.55	90.08	91.71	92.02	93.18	99.04	91.92

Table 8 presents the prediction results for the test data using the Decision Tree model with the Ensemble Extra Trees. In the table above, the resampled data using SMOTENN with 50 trees produced the best average values for accuracy, precision, recall, F1 Score, and ROC-AUC.

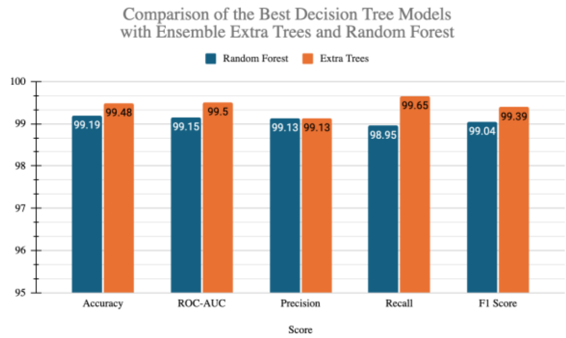

Figure 8. Comparison of the Best Decision Tree Models with Ensemble Extra Trees and Random Forest

The comparison of the best models from each algorithm, between Decision Tree with Ensemble Extra Trees and Random Forest, shows that the Decision Tree with Ensemble Extra Trees achieved the best average values. It produced an accuracy of 99.48%, ROC-AUC of 99.50%, precision of 99.13%, recall of 99.65%, and an F1 Score of 99.39%.

3.3.2. Deep Learning

3.3.2.1. Keras Tuner for Hyperparameter Search in Models and Deep Learning Algorithms

The use of Keras Tuner will be applied in selecting Deep Learning architecture to optimize the chosen hyperparameters. Hyperparameter selection is performed for both model hyperparameters and algorithm hyperparameters. This selection is carried out for each result of resampled data, whether using oversampling, undersampling, or a combination of both.

Table 9. Hyperparameter Selection

Hyperparameter	**Description**
Num Layer	1 s.d 30
Num Neuron	32 s.d 512
Activation Function	Relu (Re), SELU (SE), Sigmoid (Si)
Learning Rate	0.01, 0.001, 0.0001
Jumlah Epoch	100
Batch size	50

Table 9 describes the hyperparameter values that will be used to find the best model for each resampled dataset. The resulting model will fall within the range of the specified values.

Table 10. Keras Tuner Hyperparameter Results

Parameter	Dataset Resampling Methods						
	RUS	SMOTEENN	SMOTE	ROS	SMOTETomek	ADASYN	BLSMOTE
Num Lay	18	12	10	4	25	29	27
L Rate	0.001	0.0001	0.001	0.0001	0.0001	0.001	0.001
Layer 1	288 Re	256 SE	160 Re	416 SE	192 SE	384 SI	224 Re
Layer 2	64 SI	256 SE	32 Re	32 Re	512 Re	224 SE	480 SE
Layer 3	224 SI	448 SE	32 Re	32 Re	352 SI	224 SE	480 SE
Layer 4	448 Re	416 Re	32 Re	32 Re	192 Re	256 SE	320 SE
Layer 5	224 SI	96 SE	32 Re		32 SE	96 Re	256 SE

Layer 6	416 SE	192 Re	32 Re		512 Re	288 Re	192 SE
Layer 7	32 Re	256 SI	32 Re		320 SI	352 SE	64 Re
Layer 8	448 Re	320 SI	32 Re		416 SI	128 SE	512 Re
Layer 9	352 SE	320 SE	32 Re		32 SE	96 SI	320 Re
Layer 10	288 Re	192 SE	32 Re		384 SE	320 SE	128 SE
Layer 11	224 SI	32 SI			224 SI	352 SE	384 SE
Layer 12	224 SE	352 SI			96 SE	96 Re	128 Re
Layer 13	448 Re				416 SI	512 Re	192 SI
Layer 14	128 Re				480 Re	256 SE	320 SI
Layer 15	32 Re				416 SE	224 Re	96 SI
Layer 16	160 SI				96 SI	32 SE	32 Re
Layer 17	480 SE				320 SI	96 SE	192 SI
Layer 18	288 Re				384 SI	288 SI	352 SE
Layer 19					416 Re	64 SE	256 Re
Layer 20					32 SE	64 SE	256 SI
Layer 21					416 Re	192 SE	256 SE
Layer 22					32 Re	288 Re	416 SE
Layer 23					416 Re	128 Re	96 SE
Layer 24					128 SI	416 Re	224 SI
Layer 25					64 SE	320 SE	224 SI
Layer 26						128 Re	128 SE
Layer 27						192 Re	288 SI
Layer 28						320 Re	
Layer 29						96 Re	

Table 10 presents the most optimal model results using Keras Tuner, which provides recommendations on the number of layers, the number of neurons per layer, the activation function for each layer, and the most optimal learning rate based on daily resampled data. The most optimal learning rate across all resampling methods is 0.001. The optimal number of layers varies for each resampled dataset, as shown in Table 10. The activation function used for each layer also differs. The model results above are compiled using the Adam optimizer, binary cross-entropy as the loss function, and accuracy as the evaluation metric.

Table 11. Prediction Results of Testing Data After Hyperparameter Optimization Using Keras Tuner

PARAMETER	RUS	SMOTEENN	SMOTE	ROS	SMOTETomek	ADASYN	BLSMOTE
Accuracy	89.05%	95.82%	91.49%	90.38%	90.75%	89.91%	91.79%

Table 11 presents the prediction results of the testing data in the DNN model after hyperparameter optimization using Keras Tuner. In the table above, the resampled data using SMOTE ENN achieves the highest accuracy.

3.3.2.2. Optimizer Selection

The optimizers used are Stochastic Gradient Descent (SGD) and Adaptive Moment Estimation (Adam). Using the resampled data from SMOTEENN and the selected model from Keras Tuner tuning, the most optimal optimizer will be applied for processing. For this testing process, 200 epochs are set.

Table 12. Prediction Results of Adam and SGD Optimizers

PARAMETER	ADAM	SGD
Accuracy	96.30%	93.72%
Precision	99.15%	92.76%

Recall	92.24%	93.01%
F1	95.54%	93.01%
Cohen's Kappa	92.37%	87.16%

Table 12 and Figure 9, the results show that using the Adam optimizer yields better performance compared to the SGD optimizer across all parameters, including accuracy, precision, recall, F1 Score, and Cohen's Kappa.

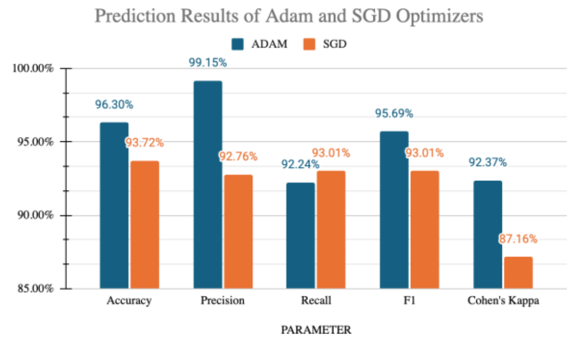

Figure 9. Prediction Results of Adam and SGD Optimizers

A more detailed evaluation of the model results using the Adam optimizer is presented in Figure 10.

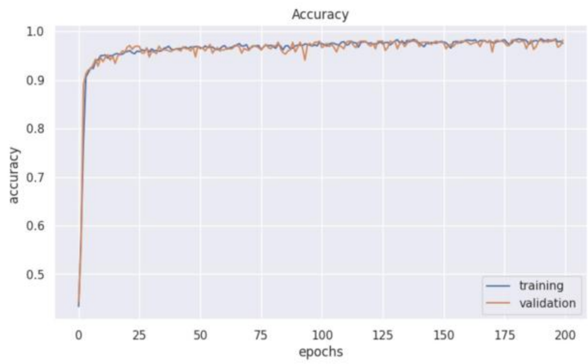

Figure 10. Model Accuracy with Adam Optimizer at Epoch 200

Figure 10 shows that at epoch 200, the accuracy of both the training and validation data continues to increase. However, in some epochs, fluctuations in accuracy can be observed. To further optimize the model, the next step will involve applying dropout and early stopping to improve performance.

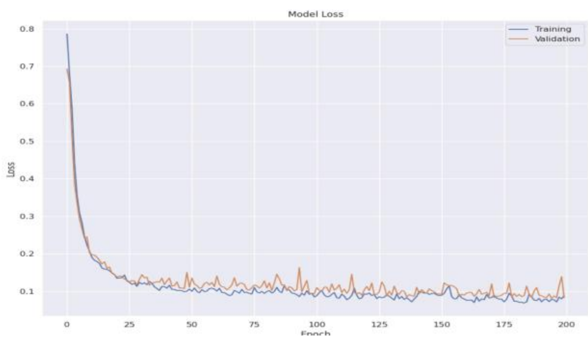

Figure 11. Model Loss with Adam Optimizer at Epoch 200

Figure 11 shows that at epoch 200, the loss value for both training and validation data continues to fluctuate. However, in some epochs, an upward trend in loss can be observed. To further optimize the model, the next step will involve applying dropout and early stopping to enhance performance.

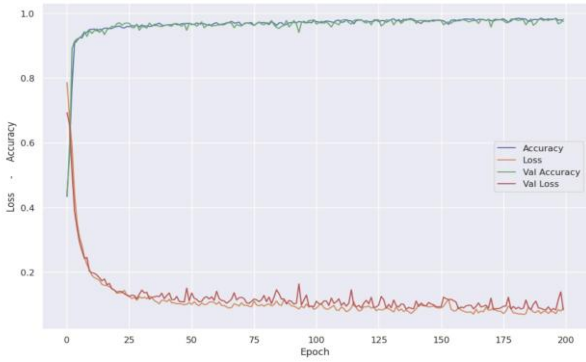

Figure 12. Model Loss and Accuracy with Adam Optimizer at Epoch 200

Figure 12 shows that at epoch 200, there is an alignment between the loss and accuracy values. As the loss decreases, the accuracy increases.

3.3.2.3. Penggunaan Dropout dan Early Stopping

The early stopping applied is set to 20 epochs. If no decrease in loss is observed within 20 consecutive epochs, the training process will be stopped.

Table 13. Prediction Results of Adam Optimizer with Dropout and Early Stopping

PARAMETER	OPTIMIZER ADAM
Accuracy	96.85%
Precision	99.15%
Recall	92.67%
F1	96.20%
Cohen's Kappa	93.51%

Table 13 presents the prediction results using the Adam optimizer with the addition of dropout in the input layer and early stopping set to 20 epochs. Early stopping occurred at epoch 116, indicating that after this point, no further decrease in loss was observed, leading to the termination of the training process.

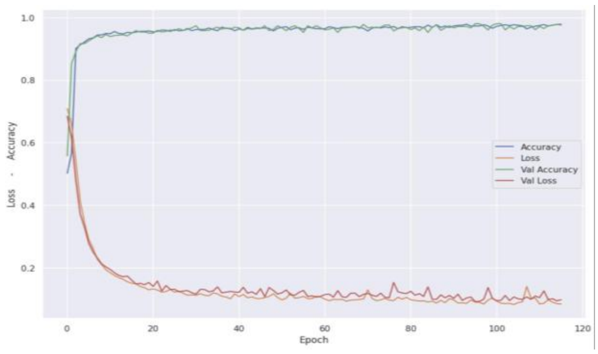

Figure 13. Loss and Accuracy Values with Dropout and Early Stopping

Figure 13 illustrates that at epoch 116, the training process was stopped even though the total epochs were set to 200. This approach optimizes time efficiency while achieving the best accuracy for the model.

3.3.2.4. Model Performance Evaluation Matrix

Table 14 is an image displaying the confusion matrix of the developed model.

Table 14. Performance Metrics and Confusion Matrix

	Predicted		
Actual	**Actual**	**True**	**False**
	True	306 TP	2 FP
	False	18 FN	214 TN

Table 14 represents the Performance Metrics, which include Accuracy, Precision, Recall, F1 Score, and Cohen's Kappa. With an accuracy of 96.85%, the model demonstrates its ability to accurately predict patients diagnosed with diabetes mellitus. The precision value indicates that the model can predict with 99.15% accuracy among all patients identified as having diabetes mellitus. The recall value shows that the model can predict with 92.67% accuracy among all patients who actually have diabetes mellitus. The F1 Score, which is the harmonic mean of precision and recall, achieves an accuracy of 96.20%. The model has a Cohen's Kappa value of 93.51%, indicating a significant agreement between variables for predicting diabetes mellitus.

3.3.2.5. Predictive Architecture Results Using DNN

Based on the results of this study, a deep learning model can be proposed to predict Diabetes Mellitus.

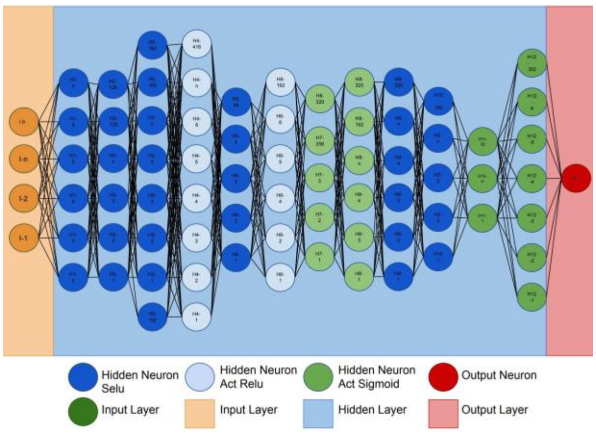

Figure 14. Design of DNN for Diabetes Mellitus Prediction

Figure 14 illustrates the proposed deep neural network model design for predicting diabetes mellitus. The parameters used in this deep neural network model are detailed in Table 15.

Table 15. Configuration of DNN for Diabetes Mellitus Prediction

Parameter	Description
Input Layer	9 *Node*
Hidden Layer	12 *Layer*
Number of Neurons in the 1st Layer and Activation Function	256 (SELU)
Number of Neurons in the 2nd Layer and Activation Function	256 (SELU)
Number of Neurons in the 3rd Layer and Activation Function	448 (SELU)
Number of Neurons in the 4th Layer and Activation Function	416 (ReLU)
Number of Neurons in the 5th Layer and Activation Function	96 (SELU)
Number of Neurons in the 6th Layer and Activation Function	192 (ReLU)
Number of Neurons in the 7th Layer and Activation Function	256 (Sigmoid)

Number of Neurons in the 8th Layer and Activation Function	320 (Sigmoid)
Number of Neurons in the 9th Layer and Activation Function	320 (SELU)
Number of Neurons in the 10th Layer and Activation Function	192 (SELU)
Number of Neurons in the 11th Layer and Activation Function	32 (Sigmoid)
Number of Neurons in the 12th Layer and Activation Function	352 (Sigmoid)
Output Layer	1
Activation Output Layer	Sigmoid
Learning Rate	0.0001
Batch size	50
Loss Function	Binary Crossentropy
Drop Out	*Input Layer*
Drop Out Rate	0.2
Optimizer	Adam
Number *Epoch*	116

The best models used in this study, Decision Tree with Ensemble Random Forest, Decision Tree with Ensemble Extra Trees, and Deep Neural Network, achieved the values as shown in Table 16.

Table 16. Best Model Performance Scores

Model	*Accuracy*	*Precision*	*Recall*	F1
Decision Tree with Ensemble Random Forest	99.19%	99.13%	98.95%	99.04%
Decision Tree with Ensemble Extra Trees	99.48%	99.13%	99.65%	99.39%
Deep Neural Network	96.85%	99.15%	92.67%	96.20%

In diabetes detection, recall is the most important metric. False Negatives (FN) must be minimized because if a person truly has diabetes but the model incorrectly predicts them as negative, they may not receive the necessary treatment. If the model misclassifies a diabetic patient as healthy, they might ignore the symptoms and face serious complications. Therefore, improving recall means detecting as many actual diabetes cases as possible. In this study, Decision Tree with Ensemble Extra Trees achieved the highest recall score of 99.65%, demonstrating its ability to detect almost all actual diabetes cases.

If a model has many False Positives (FP), it means that many healthy individuals are classified as having diabetes. This can lead to anxiety, unnecessary medical tests, or additional costs. If a healthy person is predicted to have diabetes, they might undergo tests or treatments that are not needed. Deep Neural Network achieved the highest Precision score of 99.15%, indicating its ability to accurately identify actual diabetes cases while minimizing false positives.

We can use the F1 score to maintain a balance between recall and precision. Decision Tree with Ensemble Extra Trees achieved the highest F1 score of 99.39%, indicating its strong overall performance in diabetes detection.

CONCLUSIONS

The dataset from Dr. Kariadi Central General Hospital Semarang is highly imbalanced, with 91.49% of patients classified as non-diabetic, establishing this percentage as the baseline accuracy for model development. To address this inherent imbalance in diabetes mellitus data, the SMOTEENN (*Synthetic Minority Over-sampling Technique Edited Nearest Neighbors*) method was applied, resulting in the most optimal dataset balance. Among the tested models, the Decision Tree with Ensemble Extra Trees achieved the highest accuracy at 99.48% and the highest recall at 99.65%, highlighting its effectiveness in minimizing False Negatives (FN) and ensuring that nearly all actual diabetes cases are detected—an essential aspect in clinical settings to prevent undiagnosed patients from missing timely treatment. The integration of Artificial Intelligence, particularly Decision Tree with Ensemble Extra Trees and Deep Neural Network models enhanced by SMOTEENN, has proven highly effective for diabetes prediction. For future research, it is recommended to explore advanced deep learning architectures, such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM), to further improve predictive performance and capture temporal patterns in patient data.

REFERENCES

Abdulazeez Mousa, Waraz Mustafa, Ridwan Boya Marqas, & Shivan H. M. Mohammed. (2023). A comparative study of diabetes detection using the Pima Indian diabetes database. *The Journal of The University of Duhok, 26*(2), 277-288. https://doi.org/10.26682/suod.2023.26.2.24

Achmad Fauzi, & Isnawati. (2023). Factors affecting the events of mortality in diabetes mellitus patients with complications in the ICU of Pelabuhan Jakarta Hospital. *Jurnal Ilmiah Keperawatan (Scientific Journal of Nursing), 9*(4). https://doi.org/10.33023/jikep.v9i4.1609

Aditya Budi Prasetyo, & Tri Ginanjar Laksana. (2022). Optimasi algoritma K-Nearest Neighbors dengan teknik cross validation dengan Streamlit (Studi Data: Penyakit Diabetes). Retrieved from https://jurnal.polibatam.ac.id/index.php/JAIC/article/download/4182/1835

Alex, S., Jhanjhi, N., Humayun, M., Osman, A., & Abulfaraj, A. (2022). Deep LSTM model for diabetes prediction with class balancing by SMOTE. *Electronics, 11*(17). https://doi.org/10.3390/electronics11172737

Ariani Indrawati. (2021). Penerapan teknik kombinasi oversampling dan undersampling untuk mengatasi permasalahan imbalanced dataset. *JIKO (Jurnal Informatika dan Komputer), 4*(1), 38-43. https://doi.org/10.33387/jiko.2021

Bala Manoj Kumar P., Srinivasa Perumal R., Nadesh R. K., & Arivuselvan K. (2020). Type 2 diabetes mellitus prediction using deep neural networks classifier. *International Journal of Cognitive Computing in Engineering, 1*, 55-61. https://doi.org/10.1016/j.ijcce.2020.10.002

Buffum Taylor, R., & Gatta, F. (n.d.). Types of diabetes mellitus. Retrieved from https://www.webmd.com/diabetes/types-of-diabetes-mellitus

Christian Garbin, Xingquan Zhu, & Oge Marques. (2020). Dropout vs. batch normalization: An empirical study of their impact to deep learning. *Multimedia Tools and Applications, 79*, 12777–12815. https://doi.org/10.1007/s11042-019-08453-9

Fregoso-Aparicio, L., Noguez, J., Montesinos, L., & García-García, J. (2021). Machine learning and deep learning predictive models for type 2 diabetes: A systematic review. *Diabetology & Metabolic Syndrome, 13*, Article 148. https://doi.org/10.1186/s13098-021-00767-9

García-Ordás, M. T., Benavides, C., Benítez-Andrades, J. A., Alaiz-Moretón, H., & García-Rodríguez, I. (2021). Diabetes detection using deep learning techniques with oversampling and feature augmentation. *Comput Methods Programs Biomed, 198*, 105968. https://doi.org/10.1016/j.cmpb.2021.105968

Jason Brownlee. (n.d.). Random oversampling and undersampling for imbalanced classification. Retrieved from https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/

L. Abhishek. (2020). Optical character recognition using ensemble of SVM, MLP and extra trees classifier. In *International Conference for Emerging Technology (INCET)* (pp. 1-4). https://doi.org/10.1109/INCET49848.2020.9154050

Muhammad Zahir, & Rizal Adi Saputra. (2024). Deteksi penyakit retinopati diabetes menggunakan citra mata dengan implementasi deep learning CNN. *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), 9*(3). https://doi.org/10.29100/jipi.v9i3.5588

Rahman, M., Islam, D., Mukti, R. J., & Saha, I. (2020). A deep learning approach based on convolutional LSTM for detecting diabetes. *Computational Biology and Chemistry, 88*, 107329. https://doi.org/10.1016/j.compbiolchem.2020.107329

Sharma, T., & Shah, M. (2021). A comprehensive review of machine learning techniques on diabetes detection. *Vis. Comput. Ind. Biomed. Art, 4*, 30. https://doi.org/10.1186/s42492-021-00097-7

Sunge, A., Spits Warnars, H. L. H., Heryadi, Y., Abdurachman, E., Soewito, B., & Gaol, F. (2019). Prediction diabetes mellitus using decision tree models. *2019*. https://doi.org/10.1109/AIT49014.2019.9144971

Wee, B. F., Sivakumar, S., Lim, K. H., et al. (2024). Diabetes detection based on machine learning and deep learning approaches. *Volume 83*, 24153–24185. https://doi.org/10.1007/s11042-023-16407-5

Z. P. Agusta, & Adiwijaya. (2019). Modified balanced random forest for improving imbalanced data prediction. *Int. J. Adv. Intell. Informatics, 5*(1), 58–65. https://doi.org/10.26555/ijain.v5i1.255

Purbolaksono, M. D., Irvan Tantowi, M., Imam Hidayat, A., & Adiwijaya, A. (2021). Perbandingan support vector machine dan modified balanced random forest dalam deteksi pasien penyakit diabetes. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 5*(2), 393-399. https://doi.org/10.29207/resti.v5i2.3008