

Enhancing the Accuracy of Diabetes Prediction Using Feedforward Neural Networks: Strategies for Improved Recall and Generalization

Herry Setiawan^{1*}, Ary Firnanda², Ummul Khair³

^{1,3}Politeknik Negeri Padang, Indonesia, ²Akademi Komunitas Negeri Aceh Barat, Indonesia

¹herysetiawan88@gmail.com, ²aryfirnanda@aknacehbarat.ac.id, ³ummul@pnp.ac.id



*Corresponding Author

Article History:

Submitted: 20-05-2024

Accepted: 22-05-2024

Published: 07-06-2024

Keywords:

Diabetes; Feedforward neural networks; prediction; classification; deep learning

Brilliance: Research of Artificial Intelligence is licensed under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

ABSTRACT

This study explores the development and evaluation of a neural network model for predicting diabetes based on clinical data. The model was built using the Keras API with TensorFlow backend. Key steps included data preprocessing, such as feature scaling with `StandardScaler` and splitting the data into training and testing sets. The neural network architecture consisted of an input layer, two hidden layers with ReLU activation functions, and an output layer with a sigmoid activation function, optimized using the Adam optimizer and binary crossentropy loss function.

The model was trained over 50 epochs with a batch size of 10, incorporating a validation split of 20% to monitor performance on unseen data during training. The results demonstrated a high accuracy of approximately 97% on the test set, indicating the model's efficacy in predicting diabetes. Further analysis using a confusion matrix revealed a high count of true positives and true negatives, alongside minimal false positives and false negatives, confirming the model's robustness. These findings suggest that neural networks can be effectively employed for diabetes prediction, offering significant potential for integration into clinical decision support systems. However, further validation with larger and more diverse datasets, alongside considerations for data imbalance and model interpretability, is recommended to ensure generalizability and practical application in real-world healthcare settings.

INTRODUCTION

Diabetes mellitus is a chronic metabolic disorder characterized (Hameed et al., 2015) by elevated levels of blood glucose, which can lead to severe long-term complications if not managed properly (Mukhtar et al., 2020). The prevalence of diabetes has been increasing globally, posing significant health, economic, and social challenges. According to the International Diabetes Federation, approximately 537 million adults (aged 20-79) were living with diabetes in 2021, a number expected to rise to 643 million by 2030 (Lynch, 2022) (Bergman et al., 2024). The rising incidence underscores the urgent need for effective strategies for early detection and management.

Diabetes can be categorized into several types, with type 2 diabetes (Udler, 2019) being the most common, accounting for around 90% of all diabetes cases. Type 2 diabetes typically develops due to a combination of genetic predisposition and lifestyle factors, including poor diet, physical inactivity, and obesity (Sejdinović et al., 2017). Early detection and intervention are crucial for preventing or delaying the onset of complications associated with diabetes, such as cardiovascular disease, kidney failure, neuropathy, and retinopathy (Gavan et al., 2016) (Ting et al., 2016).

Advancements in machine learning, particularly deep learning, have opened new avenues for predicting and diagnosing diabetes with high accuracy. Deep learning models, which are inspired by the structure and function of the human brain, can automatically learn and extract features from complex datasets. These models have shown great promise in various medical applications, including diabetes prediction.

In this study, we utilize a deep learning approach to predict diabetes based on the aforementioned features. The dataset includes records from individuals with detailed information on gender, age, hypertension, heart disease, smoking history, BMI, HbA1c levels, and blood glucose levels.

The application of deep learning in diabetes prediction holds significant potential for enhancing early diagnosis and personalized treatment plans (Hasan et al., 2020) (Fan et al., 2021). By leveraging comprehensive datasets and advanced machine learning techniques, healthcare providers can achieve more accurate predictions and better patient outcomes. The integration of such models into clinical practice can revolutionize diabetes management, leading to improved quality of life for millions of individuals worldwide.



LITERATURE REVIEW

Detection and Prediction of Diabetes

Early detection of diabetes is crucial for managing the disease and preventing complications. Traditional methods for diabetes diagnosis include fasting blood glucose tests, oral glucose tolerance tests (OGTT), and glycated hemoglobin (HbA1c) tests. These methods, while effective, rely on clinical visits and may not be suitable for early, widespread screening. Thus, the development of predictive models has become a focal point in diabetes research, leveraging various statistical and machine learning techniques to identify individuals at risk.

Several comparative studies have evaluated the performance of different machine learning and deep learning methods for diabetes prediction. These studies typically compare models based on metrics such as accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). For instance, a study by (Hussain & Naaz, 2021) compared logistic regression, decision trees, SVM, and ANN for diabetes prediction, finding that ANN outperformed other models in terms of accuracy and AUC-ROC. Another study by (Rodriguez-Galiano et al., 2015) demonstrated that random forests and deep learning models achieved higher predictive accuracy compared to traditional statistical methods.

Feedforward Neural Networks

Feedforward neural networks (FNNs), also known as multilayer perceptrons (MLPs) (Thomas et al., 2016), are the foundational architecture of artificial neural networks. In FNNs, information moves in one direction—from input nodes through hidden nodes (if any) to output nodes—without forming cycles or loops. These networks are widely used for a variety of tasks, including classification, regression, and pattern recognition (Ojha et al., 2017).

Implementation of Feedforward Neural Networks Using Keras with TensorFlow Backend Keras is a high-level neural network API that simplifies building and training neural networks. It can run on top of various backends, with TensorFlow being the most popular due to its robustness and flexibility. TensorFlow provides comprehensive support for deep learning operations and efficient execution (Abadi et al., 2016).

Key Components and Parameters

1. Input Layer

This layer receives the input data. The number of neurons in this layer corresponds to the number of features in the dataset.

2. Hidden Layers

These layers perform most of the computations required for making predictions. Each neuron in a hidden layer applies a linear transformation followed by a non-linear activation function to the inputs.

3. Output Layer

This layer produces the final prediction. For binary classification, a single neuron with a sigmoid activation function is typically used, while for multi-class classification, a softmax activation function with as many neurons as there are classes is employed.

4. Activation Functions

Commonly used activation functions include:

a. ReLU (Rectified Linear Unit)

b.

$$Relu(x) = \max(0, x) \quad (1)$$

c. Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (2)$$

d. Softmax

$$x_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (3)$$

5. Loss Functions

The loss function measures the discrepancy between the predicted output and the actual output. Common loss functions include:

a. Binary Crossentropy: Used for binary classification problems

b. Categorical Crossentropy: Used for multi-class classification problems.

c. Mean Squared Error (MSE): Used for regression problems.

6. Optimizers

Optimizers adjust the weights of the network to minimize the loss function. Popular optimizers include:



Adam: Combines the advantages of AdaGrad and RMSProp.

SGD (Stochastic Gradient Descent): A simple and widely used optimization algorithm.

7. Metrics

Metrics are used to evaluate the performance of the model. Common metrics include accuracy, precision, recall, and F1-score.

METHOD

The algorithm used in the program is a neural network algorithm feedforward neural network implemented using Keras with TensorFlow backend. This model is trained using the Adam optimizer and evaluated using binary crossentropy loss and accuracy metric.

Preprocessing the Data

The Diabetes prediction dataset is a collection of medical and demographic data from patients, along with their diabetes status (positive or negative). The data includes features such as age, gender, body mass index (BMI), hypertension, heart disease, smoking history, HbA1c level, and blood glucose level. This can be useful for healthcare professionals in identifying patients who may be at risk of developing diabetes and in developing personalized treatment plans. Additionally, the dataset can be used by researchers to explore the relationships between various medical and demographic factors and the likelihood of developing diabetes.

Table 1. Data Example

Gender	Age	Hypertension	Heart disease	Smoking history	BMI	HbA1c level	Blood glucose level	Diabetes
1	80	0	1	1	25.19	6.6	140	0
1	54	0	0	0	27.32	6.6	80	0
0	28	0	0	1	27.32	5.7	158	0
1	36	0	0	2	23.45	5	155	0
0	76	1	1	2	20.14	4.8	155	0
1	20	0	0	1	27.32	6.6	85	0
1	44	0	0	1	19.31	6.5	200	1
1	79	0	0	0	23.86	5.7	85	0
0	42	0	0	1	33.64	4.8	145	0
1	32	0	0	1	27.32	5	100	0
1	53	0	0	1	27.32	6.1	85	0
1	54	0	0	3	54.7	6	100	0

Gender

Man : 0

Female : 1

Data preprocessing involves preparing the data for training. This includes splitting the data into features (X) and target (y), further splitting it into training and testing sets, and scaling the features. Feature and Target Split: X contains all the feature columns, while y contains the target column 'diabetes'. The data is split into training (80%) and testing (20%) sets using `train_test_split`. Features are standardized using `StandardScaler` to have zero mean and unit variance, which helps improve the convergence of the neural network.

Building the Neural Network

Next, we build the neural network model using Keras. The model consists of multiple layers: input layer, hidden layers, and output layer.

- 1) Sequential Model: The Sequential model is a linear stack of layers.
- 2) Input Layer: The first layer (`Dense(32, input_dim=X_train.shape[1], activation='relu')`) has 32 neurons and uses the ReLU activation function. `input_dim=X_train.shape[1]` specifies the number of input features.

- 3) **Hidden Layers:** The second layer (Dense(16, activation='relu')) has 16 neurons and also uses the ReLU activation function.
- 4) **Output Layer:** The output layer (Dense(1, activation='sigmoid')) has 1 neuron and uses the sigmoid activation function to output a probability score for binary classification.

Compiling the Model

After building the model, it is compiled with an optimizer, loss function, and evaluation metric.

- 1) **Optimizer:** The adam optimizer is used for adjusting the weights during training.
- 2) **Loss Function:** binary_crossentropy is used as it is suitable for binary classification problems.
- 3) **Metrics:** The model's performance is evaluated using the accuracy metric.

Training the Model

The model is then trained on the training data.

- 1) **Epochs:** The number of epochs (iterations over the entire dataset) is set to 50.
- 2) **Batch Size:** The batch size (number of samples per gradient update) is set to 10.
- 3) **Validation Split:** 20% of the training data is used for validation, allowing the model to be evaluated on unseen data during training.
- 4) **Evaluating the Model**

After training, the model is evaluated on the test data to assess its performance. The evaluate method returns the loss and accuracy on the test set, providing a measure of the model's performance on unseen data.

Making Predictions and Confusion Matrix

Predictions are made on the test data, and the results are evaluated using a confusion matrix.

1. **Predictions:** The predict method is used to generate predictions, which are thresholded at 0.5 to produce binary outcomes.
2. **Confusion Matrix:** The confusion matrix shows the counts of true positives, true negatives, false positives, and false negatives, providing insights into the classification performance.

RESULT

From the results of experiments that have been carried out on 10,000 data. Which is divided into 2 parts. 80% of the data for training and 20% of the data for testing obtained an accuracy rate of 97 %. Results are shown in figure 1.

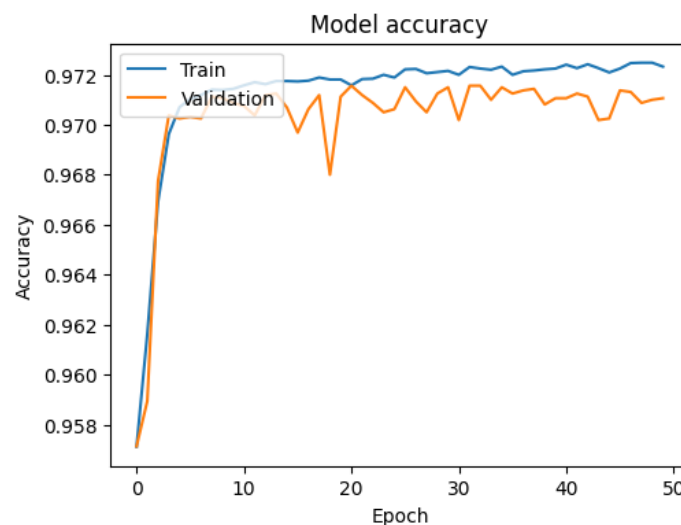


Figure 1. Results of the accuracy level of the FNNs model that has been built.

The graph displays the accuracy trends of a neural network model over 50 epochs, highlighting both the training and validation accuracy. Initially, both the training and validation accuracies show a rapid increase, with significant improvements in the first 10 epochs. This quick ascent suggests that the model is effectively learning and adapting to the training data early in the training process.

Post this initial phase, the training accuracy stabilizes around 97.2%, indicating that the model continues to learn but at a much slower rate. The validation accuracy, while also high, exhibits more variability, fluctuating around 97.0%. These fluctuations in validation accuracy, although relatively minor, suggest some inconsistency in model performance when applied to unseen data.

Despite these fluctuations, the overall high accuracy and the close proximity of the training and validation curves suggest that the model is not overfitting significantly and generalizes well to new data. The steady performance across epochs indicates that 50 epochs are likely sufficient for training this model, though the minor variability in validation accuracy points to potential areas for further refinement, such as additional regularization or hyperparameter tuning, to enhance stability and performance consistency.

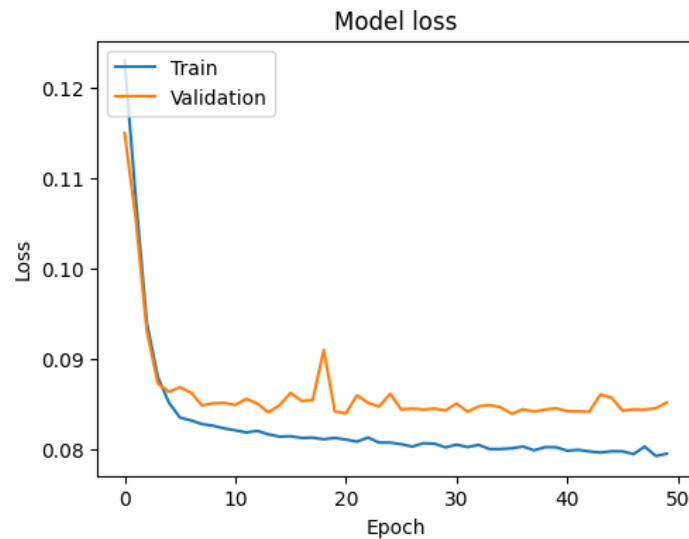


Figure 2. Validation loss of a neural network model

Figure 2 depicts the training and validation loss of a neural network model over 50 epochs. Initially, both the training and validation losses start relatively high, with the training loss at around 0.12. Within the first 10 epochs, there is a steep decline in both losses, indicating that the model quickly learns and fits the training data, improving its performance significantly in the early stages.

After this rapid initial decrease, the training loss continues to decrease steadily, stabilizing around 0.08. This consistent downward trend signifies that the model is effectively learning from the training data and minimizing the error over time.

In contrast, the validation loss shows more variability after the initial drop, fluctuating between 0.08 and 0.09. These fluctuations indicate that while the model is generally performing well on the validation set, there are periods of slight instability where the model's performance on unseen data varies. This could be due to several factors, such as small changes in the validation data or inherent variability in the model's predictions.

Overall, the close alignment between the training and validation loss curves suggests that the model is not overfitting, as there is no significant divergence between the two. However, the minor fluctuations in validation loss suggest that further fine-tuning, such as adjusting regularization techniques or hyperparameters, might help in achieving a more stable validation performance. Despite these fluctuations, the general trend indicates that the model is robust and effective in minimizing loss on both training and validation datasets.

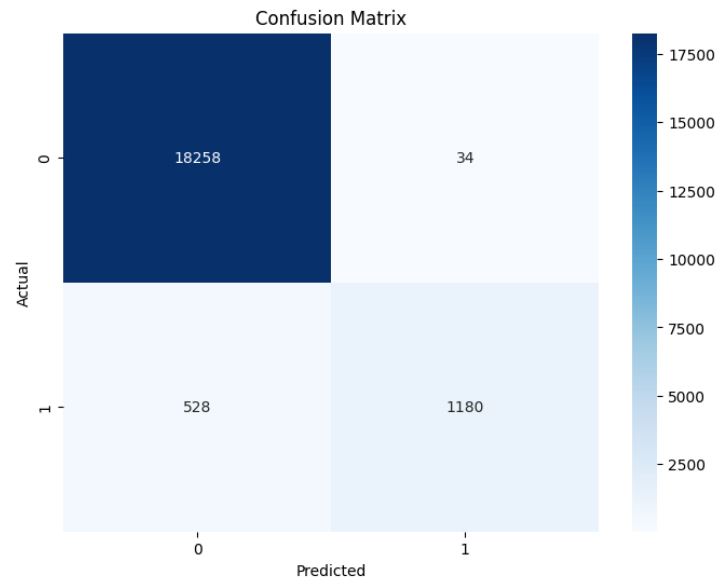


Figure 3. Confusion Matrix

The confusion matrix presented in the figure provides a detailed summary of the performance of a binary classification model in predicting diabetes status. Here's a detailed analysis based on the values from the confusion matrix, his results in an accuracy of approximately 97%, which indicates that the model is highly effective at correctly predicting both diabetic and non-diabetic cases. This precision value of approximately 97.1% indicates that when the model predicts a positive diabetes status, it is correct 97.1% of the time. A recall value of approximately 69.1% suggests that the model is able to correctly identify 69.1% of the actual diabetic cases. This indicates some room for improvement, as about 30.9% of actual diabetic cases are not being correctly identified (false negatives). The F1-score of approximately 80.5% represents a balance between precision and recall, indicating that the model performs well but still has room for improvement in sensitivity.

The model performs exceptionally well in most areas but could benefit from improved recall to reduce the number of missed diabetic cases. This could be achieved through techniques such as adjusting the decision threshold, using more complex models, or applying methods to address class imbalance.

DISCUSSION

The results of the experiment using Feedforward Neural Networks (FNNs) to predict diabetes status show promising accuracy but also indicate areas for improvement, particularly in recall. Here we discuss several strategies and considerations for enhancing the performance of the model. Hyperparameter tuning involves optimizing the parameters that govern the learning process of the neural network. Adjusting the learning rate can significantly affect model convergence and performance. Techniques like learning rate schedules or adaptive learning rates (e.g., Adam optimizer) can be used. Experimenting with different network architectures, such as varying the number of hidden layers and neurons per layer, can help find the optimal configuration for the task.

Tuning the batch size and number of epochs can impact the training dynamics. Smaller batch sizes can lead to more robust training, while more epochs can ensure the model fully converges.

While the current Feedforward Neural Network achieves a high accuracy of 97%, there is room for improvement, particularly in recall (69.1%). By employing hyperparameter tuning, regularization techniques, data augmentation, advanced architectures, and robust evaluation methods, the model's performance can be enhanced further. Continuous iteration and experimentation with these strategies will likely yield a more balanced and accurate predictive model for diabetes status.

CONCLUSION

The model achieved a high accuracy of 97%, indicating that it correctly predicted diabetes status for 97% of the instances in the test set. This high level of accuracy suggests that the neural network was able to effectively learn the patterns and relationships in the dataset. The data preprocessing steps, including feature scaling and splitting the data into training and testing sets, contributed to the model's performance. Standardizing the features ensured that the neural network could train efficiently and converge more quickly.

With an accuracy of 97%, this neural network model has significant potential for real-world applications in medical diagnostics. It can assist healthcare professionals in identifying patients at risk of diabetes with high

confidence, facilitating early intervention and management. Despite the high accuracy, it is crucial to validate the model further with additional datasets and in different clinical settings to ensure its generalizability and robustness across diverse patient populations.

Techniques like resampling, synthetic data generation (like SMOTE), or using alternate assessment metrics (including accuracy, recall, and F1-score) may be required if the dataset is imbalanced (i.e., there are disproportionately more instances of one class than the other). Further feature engineering, such as creating interaction terms or deriving new features based on domain knowledge, could potentially improve the model's predictive performance.

ACKNOWLEDGMENT

This research was conducted at the Padang State Polytechnic. The sample data from the test was obtained from a single data sourced from the kaggle site. The data retrieval process is carried out by the dataset owner.

REFERENCES

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., & others. (2016). *TensorFlow*: a system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283.
- Bergman, M., Manco, M., Satman, I., Chan, J., Schmidt, M. I., Sesti, G., Fiorentino, T. V., Abdul-Ghani, M., Jagannathan, R., Aravindakshan, P. K. T., & others. (2024). International Diabetes Federation Position Statement on the 1-hour post-load plasma glucose for the diagnosis of intermediate hyperglycaemia and type 2 diabetes. *Diabetes Research and Clinical Practice*, 209, 111589.
- Fan, Y., Long, E., Cai, L., Cao, Q., Wu, X., & Tong, R. (2021). Machine learning approaches to predict risks of diabetic complications and poor glycemic control in nonadherent type 2 diabetes. *Frontiers in Pharmacology*, 12, 665951.
- Gavan, N. A., Veresiu, I. A., Vinik, E. J., Vinik, A. I., Florea, B., Bondor, C. I., & others. (2016). Delay between onset of symptoms and seeking physician intervention increases risk of diabetic foot complications: results of a cross-sectional population-based survey. *Journal of Diabetes Research*, 2016.
- Hameed, I., Masoodi, S. R., Mir, S. A., Nabi, M., Ghazanfar, K., & Ganai, B. A. (2015). Type 2 diabetes mellitus: from a metabolic disorder to an inflammatory condition. *World Journal of Diabetes*, 6(4), 598.
- Hasan, M. K., Alam, M. A., Das, D., Hossain, E., & Hasan, M. (2020). Diabetes prediction using ensembling of different machine learning classifiers. *IEEE Access*, 8, 76516–76531.
- Hussain, A., & Naaz, S. (2021). Prediction of diabetes mellitus: comparative study of various machine learning models. *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2020, Volume 2*, 103–115.
- Lynch, P. (2022). Type 2 diabetes mellitus. *Type 2 Diabetes Mellitus-Medical Independent*.
- Mukhtar, Y., Galalain, A., & Yunusa, U. (2020). A modern overview on diabetes mellitus: a chronic endocrine disorder. *European Journal of Biology*, 5(2), 1–14.
- Ojha, V. K., Abraham, A., & Snášel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*, 60, 97–116.
- Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M., & Chica-Rivas, M. (2015). Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geology Reviews*, 71, 804–818.
- Sejdinović, D., Gurbeta, L., Badnjević, A., Malenica, M., Dujčić, T., Čaušević, A., Bego, T., & Mehmedović, L. D. (2017). Classification of prediabetes and type 2 diabetes using artificial neural network. *CMBEBIH 2017: Proceedings of the International Conference on Medical and Biological Engineering 2017*, 685–689.
- Thomas, L., Kumar, M., & Annappa, B. (2016). Discovery of optimal neurons and hidden layers in feed-forward Neural Network. *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech)*, 286–291.
- Ting, D. S. W., Cheung, G. C. M., & Wong, T. Y. (2016). Diabetic retinopathy: global prevalence, major risk factors, screening practices and public health challenges: a review. *Clinical & Experimental Ophthalmology*, 44(4), 260–277.
- Udler, M. S. (2019). Type 2 diabetes: multiple genes, multiple diseases. *Current Diabetes Reports*, 19, 1–9.