Jurnal Elektronika dan Telekomunikasi (JET), Vol. 24, No. 2, December 2024, pp. 80-87
Accredited by KEMDIKBUDRISTEK, Decree No: **158/E/KPT/2021**
doi: 10.55981/jet.640

Deep Neural Network Classifier for Analysis of the Debrecen Diabetic Retinopathy Dataset

Cucu Ika Agustyaningrum [a, *], Haryani [a], Agus Junaidi [a], Iwan Fadilah [b]

[a]Fakultas Teknik dan Informatika
Universitas Bina Sarana Informatika
Jl. Kramat Raya No.98, RT.2/RW.9, Kwitang, Senen
Jakarta, Indonesia
[b]Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh November
Jl. Teknik Kimia, Keputih, Kec. Sukolilo, Surabaya,
Jawa Timur, Indonesia

Abstract

Diabetic retinopathy (DR) is a serious complication that can occur in individuals who have diabetes. This disease affects the blood vessels in the retina, a part of the eye that is important for vision. Early detection of DR is key to preventing further complications and saving the patient's vision. The goal of Diabetic Retinopathy Debrecen Data Set Analysis is to get the best, most accurate results for medical professionals to receive appropriate Diabetic Retinopathy Debrecen prediction results through the stages of data collection, evaluation, and classification. Data is collected from existing secondary sources and then assessed using a deep neural network algorithm with various variations. The classification algorithm in this research uses the Python programming language to measure accuracy, F1-Score, precision, recall, and receiver operating characteristic (ROC) area under the curve (AUC). The test results show that the accuracy of the deep neural network algorithm is 79.94%, the F1 score reaches 79.16%, the precision is 79.58%, the recall is 79.60%, and the AUC is 79.56%. Thus, based on this research, the deep neural network data mining technique with variations of the four hidden layer encoder-decoder, sigmoid activation function, Adam optimizer, learning rate 0.001, and dropout 0.2 is proven to be effective. When compared with other variations such as decoder-encoder, 3-8 hidden layers, learning rate 0.1 and 0.01, the average difference in values between this variation and the others are 0.07% accuracy, 2.03% F1 score, 0.25% precision, 0.80% recall, and 0.90% AUC. Therefore, the deep neural network algorithm with the variation used shows significant dominance compared to other variations.

Keywords: algorithm, classification, diabetic retinopathy, deep neural network.

I. INTRODUCTION

Diabetic retinopathy (DR) is a serious complication of diabetes that affects the blood vessels in the retina, a critical part of vision. DR is one of the main causes of blindness in adults worldwide [1].

Preventing further complications and saving the patient's vision requires early detection of DR. However, manual diagnosis of DR by an ophthalmologist requires a large amount of time and expense and may be affected by human subjectivity. Ophthalmologists or medical professionals can receive assistance on how to determine patient survival using data mining techniques [2].

A procedure known as data mining combines several computer science specialists to find new patterns in enormous amounts of data. Techniques from database systems, statistics, machine learning, and artificial intelligence are used in this procedure. Devi and Rahayu [3] interpret data mining as automatically searching through data to generate models from huge databases [3].

Data mining can handle large volumes of data, often not managed by conventional methods. Data mining algorithms are designed to process large amounts of data quickly and efficiently. There are various functions, one of which is the classification function. Depending on the number of required classes, objects are grouped into many classes using the data processing approach of classification. Classification is a technique of finding a pattern that can separate one data class from another to determine which objects fall into a certain category by looking at the behavior and attributes of the groups that have been defined. This technique can clarify new data using the results to provide several rules. One technique that can be used to determine classification is deep learning [4].

In recent years, deep learning technology has demonstrated its ability to automate medical diagnosis processes, including DR detection from retinal images. This research aims to apply a deep learning approach to analyze the DR Debrecen dataset, which is a large dataset and has been used in various previous studies [5].

Previous research introduced a deep learning model that uses convolutional neural networks (CNN) to classify retinal images into five different DR classes [6]. This model achieved promising levels of accuracy and showed great potential for automatic DR detection.

These researchers developed a deep learning system that is able to identify DR with a high level of accuracy

* Corresponding Author.
Email: cucu.cgy@bsi.ac.id
Received: March 27, 2024 ; Revised: June 02, 2024
Accepted: June 06, 2024 ; Published: December 31, 2024

[7]. They used a large dataset to train their model and performed cross-institutional validation to test the generalization of the model.

Research conducted by [8] created a deep learning model that can classify retinal images as normal or experiencing DR. The model achieved accuracy that competes with human diagnosis, showing potential for use in clinical practice.

Another study by [9] focused on developing a deep learning algorithm for DR detection in retinal images taken with smartphones. They show that this technology can be integrated into broader healthcare systems to enable early detection of DR at the population level.

Emerging problems, such as limitations in the amount and diversity of DR data, can result in difficulties in training models that can generalize well across a wide range of clinical conditions. Complex deep neural network (DNN) models may require large computational resources for training and inference and can complicate the interpretation of results. Biases in the dataset, such as differences in image quality or subject recruitment criteria, can affect model performance and generalization [10]. DNN models are frequently referred to as "black boxes" since it can be challenging to comprehend the logic underlying the model's choices. This may weaken user confidence and make the clinical validation process more difficult. The integration of deep learning technologies in clinical practice requires compliance with appropriate data security standards, medical regulations, and information technology infrastructure [11].

The solution to this research problem is using deep learning technology, such as DNN, which can enable automatic detection of DR with a high level of accuracy, speed up the diagnosis process, and reduce dependence on human labor. Proper preprocessing on retinal image data, including intensity normalization, contrast enhancement, and data augmentation, can improve data quality and help models learn relevant features. Choosing a DNN architecture that suits the data and complexity of the problem and carefully tuning parameters can improve model performance and efficiency [12]. Performing cross-validation and testing on independent datasets can help measure model generalization and avoid overfitting. Building a model that can provide an interpretation of the decisions taken can increase medical practitioners' trustworthiness and adoption of the model.

As explained above, there has been no data mining research on Diabetic Retinopathy Debrecen data sets with the DNN algorithm; there is only image processing research for this dataset. Therefore, in this research, the focus is on comparing the results of variations of the DNN algorithm, which is the best way to assess the classification of Diabetic Retinopathy Debrecen data sets using the Python programming language by processing various DNN algorithms to get the best and most accurate results, medical professionals can receive Diabetic Retinopathy Debrecen prediction results.

II. METHODS

The model utilized the phases of dataset, preprocessing, feature selection, modeling, and model evaluation and was based on the Debrecen Diabetic Retinopathy research. Data collection from the UCI Irvine Machine Learning Repository site was the first step in this procedure, after which the data was transformed into preliminary data for processing. Next, the Python programming language was used to pick features. Testing using DNN methods and cross-validation was the process's next step.

A. Research Phase

Using a DNN technique, the modeling stage first preprocessed the categorical data to convert it to all numerical format before choosing which characteristics to employ in the data processing when modeling was entered for diabetic retinopathy. In order to cross-validate the preprocessed data for the classification process, it was split into training and testing sets [13]. The test data results were analyzed using the Python programming language and the DNN algorithm to produce the best accuracy and F1 score using three to eight variations of layers used on the data. This process came after processing the data from preprocessing, feature selection, modeling, and testing. An explanation of the research method step is provided in Figure 1.

B. Implementation of Research Methods

There were five steps involved in applying the Debrecen Diabetic Retinopathy Data Research Method:

1) Dataset

The Debrecen Diabetic Retinopathy study used secondary data, which was taken from researchers who had previously conducted similar research. The Debrecen Diabetic Retinopathy data has a total of 1151 attributes, including 1 class, taken from the UCI Irvine Machine Learning Repository [14] to find out data on Debrecen Diabetic Retinopathy sufferers. A categorization system with a high degree of predictability and accuracy was chosen to get around these issues and produce simpler, quicker, and more accurate results. Using versions of the DNN method with three to eight layers, this study can

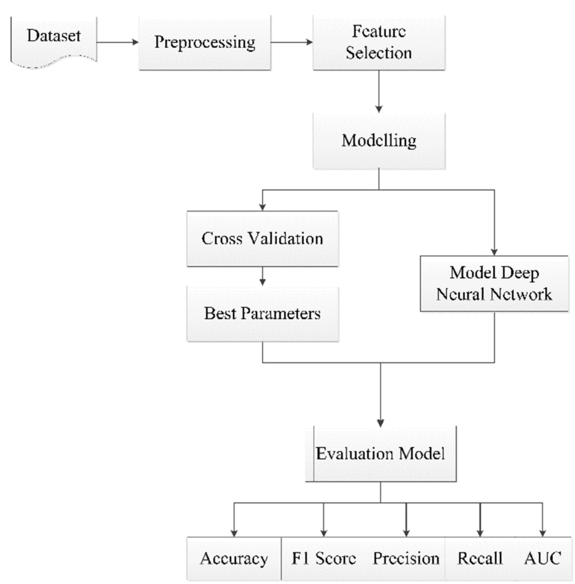

Figure 1. Research method stages.

produce precise values and high-level prediction outcomes.

2) Preprocessing

The data were analyzed and then used as a basis to identify data on diabetic retinopathy Debrecen by generating positive conditions or negative conditions for the diabetic retinopathy condition. A total of 1151 data points have been collected. Starting with the data preparation stage, the data type in the dataset attributes was then changed.

After selecting the data, the next process was to check the data for missing values; this process is called data cleaning.

3) Feature selection

Determining the features that have the greatest influence on the Debrecen diabetic retinopathy data set is the feature selection stage. After performing feature selection, the next step was to divide the data using cross-validation in the process of developing a DNN algorithm model with variations in the number of layers from 3 to 8.

4) Modeling

In the modeling stage, predictions were made using the recommended algorithm. The recommended algorithm is a DNN with variations that include the use of three to eight depth layers, various activation functions, optimizers, and learning rate levels. This process was implemented using the Python programming language to measure accuracy, F1 score, precision, recall, and area under the curve (AUC) in the classification of data on diabetic retinopathy sufferers.

a) An artificial neural network with numerous layers is called a DNN. A DNN often has more than three layers, which include an output layer, several hidden levels, and an input layer. In other words, DNN can be considered a variation of multilayer perceptron (MLP) with a larger number of layers. Due to the relatively large number of layers, this makes it a deep approach to the learning process. Deep learning is another term for the learning process in DNN [15]. The speech recognition field benefits from the deep learning approach's superior network topology, being able to optimize a large number of parameters, and DNN capabilities, which are quite good in speech recognition and speed in understanding various languages and dialects [16].

Essentially, the implementation of deep learning involves the use of various very deep layers [17]. In each layer, nodes form a smaller network to produce decisions. At each layer that has these nodes, calculations were carried out starting from input variables (such as X1, X2, X3, ...wn) [18].

Every input layer was linked to a weight, as seen in Figure 2. Equation (1) illustrates the formula that was employed [19].

$$Z = f(x.b) = f(\sum_{i=1}^{n} x_i w_i + b \quad (1)$$

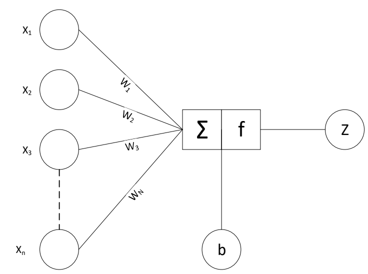

Figure 2. Illustration of the relationship between deep learning input and output.

Where *Z* is the output of the activation function, which is the final result of the neuron after calculating the given input and applying the activation function *f*,

f(x \cdot b) is the activation function applied to the linear combination of the input *x* and the bias *b*. This activation function can be a sigmoid function, rectified linear unit (ReLU), or other activation function, which determines whether the neuron is "active" or not,

Σ (Sigma) is the mathematical notation used to express summation. In this case, summing the results of each input *xix_ixi* multiplied by the *weight wiw_iwi*,

x_i is the value of the i-th input (for example, a feature in the dataset or the input value from the previous layer),

w_i is the weights or weight associated with the i-th input. This weight determines how much influence the input has on the neuron's output,

and *b* is biased, which is added to shift the summation result before it is applied to the activation function. Bias helps the model to make more flexible adjustments.

So, overall, this formula describes the output of a neuron in an artificial neural network, where each input is multiplied by its weight, then summed together with the bias, and applied to the activation function.

b) Activation Function

The activation function must satisfy several important characteristics in the context of backpropagation networks. First, the activation function must be continuous, meaning it must exist and be computable at every point. Second, the activation function must be differentiable so that its derivative can be calculated. Finally, the activation function must be monotonically non-decreasing, which means its value must not decrease as the input increases. In addition, for efficiency in calculations, it is desirable if the derivative of the activation function can be obtained easily, and the value of the derivative can be expressed in terms of the activation function itself.

For example, the sigmoid activation function has a value range between 0 and 1, as shown in Figure 3, and this function is defined as an output with specific qualities that can be created from input by using the activation function. In

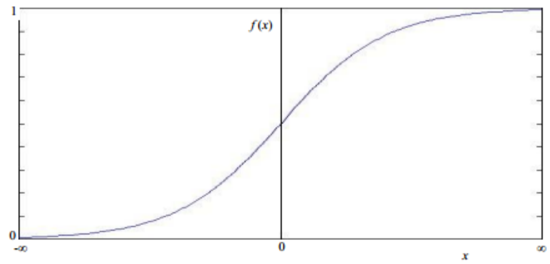

Figure 3. Sigmoid activation function.

this research, to make the output from neurons more measurable, there are two different versions of the activation function that are utilized: Sigmoid and ReLU [20], formulated in (2) and (3).

$$f(x) = \frac{1}{1+\exp(-\sigma x)} \tag{2}$$

with

$$f(x) = \sigma f(x)[1 - f(x)] \tag{3}$$

Where *f(x)* is the output of the sigmoid function, σ typically represents a constant scaling factor (often assumed to be 1 if not stated),
e is the base of the natural logarithm,
and x is the input to the function.
The sigmoid function is commonly used in neural networks to map input values into a range between 0 and 1, making it useful for binary classification.

c) *Layer*

This component in DNN involves the first layer, which functions as an input layer that receives input data. Data from the input layer was then sent to a hidden layer of the DNN for processing. Lastly, binary nodes that generate decisions throughout the classification process make up the output layer [21]. This suggests that more hidden layers lead to a longer learning process while providing the networks with more accurate targets to match inputs. However, excessive modeling capacity can lead to overfitting problems [22].

d) *Dropout*

Dropout is a technique used to prevent overfitting and can also speed up the learning process. It is important to tune the related hyperparameters to implement dropout effectively. Dropout refers to the process in which neurons in the hidden layer of a network are temporarily removed. In this context, the neurons to be deleted are randomly selected, and each neuron has a probability p between 0 and 1.0. Implementing dropout means temporarily removing these neurons from the network [23].

e) *Learning Rate*

The learning rate determines the extent of weights to be corrected throughout the training phase. The typical range for this learning rate value is zero (0) to one (1). As the learning rate value increases, the possibility of reaching the global minimum increases, but it should be noted that this can also cause non-convergence or difficulty in attaining the ideal outcome sought. Conversely, a smaller learning rate value increases the likelihood of achieving the minimum and converging. However, finding the best solution may take a while if the learning rate value is too little. Therefore, it is important to adjust hyperparameters to obtain appropriate learning rate values in model training [24].

f) *Optimizer*

Various optimizer options were tested to achieve more responsive optimization and suited to the needs [25].
An optimizer is an algorithm used to optimize various parameters in a neural network, such as weights and learning rates, to reduce loss values [26]. Four types of optimizers are stochastic gradient descent (SGD), a deep learning model that is built from the Keras model, reduces the step size, and follows gradients to optimize functions [27]. SGD is extended by Adam (Adaptive Moment Estimation) [28]. Adagrad is a method that modifies stochastic gradient descent, incorporating varying learning rates for every parameter [26]. Each parameter's learning rate is changed separately using a root mean square propagation (RMSProp) technique. The five optimizers have proven exceptionally effective in various deep-learning tasks [25].

g) *Cross Validation (K-Fold)*

A model validation technique called cross-validation is used to gauge how dependable the outcomes of the model analysis are. Following the initial data processing, the classification process applies the cross-validation technique by splitting the data into training and test sets [29]. The data partition utilized as training and test data is changed each time this test is run [30].
The model or algorithm is taught using a small portion of the learning data and then tested using another portion called validation data. The selection of the appropriate type of cross-validation can depend on the dataset size; when computing time may be decreased without compromising estimation accuracy, K-fold cross-validation is employed [31].

h) *Confusion Matrix*

Confusion matrices are an extremely helpful tool for examining discrimination in identifying components from various groups. This approach involves using an array of matrices representing the positive and negative classes. Using the confusion matrix, we can compute the accuracy, precision, recall, and error rate values during the evaluation phase. Reliability is the proportion of accurately recognized cases to the total cases [32].

The accuracy is calculated using (4) to (6).

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

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

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

Where *TP* (True Positives) is the number of instances correctly predicted as positive.

TN (True Negatives) is the number of instances correctly predicted as negative.

FP (False Positives) is the number of instances incorrectly predicted as positive (also called Type I error),

and *FN* (False Negatives) is the number of instances incorrectly predicted as negative (also called Type II error).

j) Area Under Curve (AUC)

AUC is a more holistic prediction evaluation metric when dealing with data imbalances. AUC provides an indication of classifier performance in scalar form and has been frequently used in situations where the class distribution is unbalanced. The AUC is calculated by measuring the area under the receiver operating characteristic (ROC) curve using (7) [33].

$$AUC = \left(\frac{1+TPRate-FPRate}{2}\right) \quad (7)$$

Where *TP Rate* (True Positive Rate), also known as Recall,

and FPRate (False Positive Rate) measures the proportion of actual negatives that were incorrectly predicted as positive.

5) Evaluation

In the evaluation stage, predictions are made using the DNN algorithm, with variations covering three to four front-depth layers, various activation functions, different optimizers, and varying learning rates. This step uses the Python programming language to assess the outcomes of accuracy, F1 score, precision, recall, and AUC values as a gauge of success and mistake rates.

C. Data Collection Method

Data collection methods can be grouped into two main sources: primary data and secondary data. Primary data is information obtained directly from the source, while secondary data is information obtained from research that other researchers have previously carried out in a similar context. In this study, researchers used secondary data from the Debrecen Diabetic Retinopathy Dataset obtained from the UC Irvine Machine Learning Repository. This dataset consists of 1151 records with 20 attributes and 1 class attribute, as described in Table 1.

TABLE 1
DESCRIPTION OF DEBRECEN DIABETIC RETINOPATHY DATASET ATTRIBUTES

Variable Name	Role	Type	Description	Missing Values
quality	Feature	Binary	The binary result of quality assessment. 0 = bad quality 1 = sufficient quality.	no
pre_screening	Feature	Binary	The binary result of pre-screening, where 1 indicates severe retinal abnormality and 0 its lack.	no
ma1	Feature	Integer	ma1 - ma-6 contain the results of MA detection. Each feature value stands for the number of MAs found at the confidence levels alpha = 0.5, . . . , 1, respectively.	no
ma2	Feature	Integer		no
ma3	Feature	Integer		no
ma4	Feature	Integer		no
ma5	Feature	Integer		no
ma6	Feature	Integer		no
exudate1	Feature	Continuous	exudate1 - exudate8 contain the same information as 2-7) for exudates. However, as exudates are represented by a set of points rather than the number of pixels constructing the lesions, these features are normalized by dividing the number of lesions with the diameter of the ROI to compensate for different image sizes.	no
exudate2	Feature	Continuous		no
exudate3	Feature	Continuous		no
exudate3	Feature	Continuous		no
exudate5	Feature	Continuous		no
exudate6	Feature	Continuous		no
exudate7	Feature	Continuous		no
exudate8	Feature	Continuous		no
macula_opticdisc_distance	Feature	Continuous	The Euclidean distance between the center of the macula and the center of the optic disc provides important information regarding the patient’s condition. This feature is also normalized with the diameter of the ROI.	no
opticdisc_diameter	Feature	Continuous	The diameter of the optic disc.	no
am_fm_classification	Feature	Binary	The binary result of the AM/FM-based classification.	no
Class	Target	Binary	Class label. 1 = contains signs of DR (Accumulative label for the Messidor classes 1, 2, 3), 0 = no signs of DR.	no

III. Result and Discussion

The results of this research include data processing procedures, both qualitative and quantitative, which were collected following the proposed model. This study covers all available data sets. Experiments and testing in this research involve data predictions using DNN algorithms. This experiment will be applied to a dataset that has gone through the validation stage after the preprocessing, feature selection, and modeling processes have been carried out using Google Colaboratory with the Python programming language.

In a study using the Debrecen Diabetic Retinopathy Dataset, the application of a DNN algorithm resulted in various variations in the metric values used, as described below.

A. Testing the Preprocessing Step

The next step is to convert the attributes having a variable data type to an integer data type. This conversion is done so that data mining techniques using the Python programming language may be applied once the dataset has been processed by taking into account the attributes that match Table 1. After changing the attribute's data type is complete, the next step is to clean the data. At this stage, an effort is made to check whether the data has missing values.

B. Implementation of the Deep Neural Network Algorithm Model

In the research experiments carried out, the accuracy rate in the Diabetic Retinopathy Debrecen Dataset reached 79.94% in Table 2. These results were obtained through calculations involving various variations of DNN algorithms. Specifically, this value is achieved when the DNN algorithm has 4 hidden layers with a decoder-encoder architecture, uses sigmoid function activation, an Adam optimizer with a learning rate of 0.001, and a dropout of 0.2.

Optimization involves setting parameters such as input data form, hidden layer variations, activation function, optimizer, learning rate, and dropout. Optimization is important because these five parameters significantly impact the accuracy and confusion matrix of the DNN algorithm.

The values obtained from the DNN algorithm's several model variants are displayed in Table 2. It is evident that research that has obtained acceptable accuracy levels has used the Debrecen Diabetic Retinopathy Dataset. Only a small amount separates the lowest accuracy result from the greatest accuracy figure, which is 79.94%. The following outcomes of modifying the model to include an activation function and four hidden layers in an encoder-decoder architecture were obtained: AUC was 79.56%, recall was 79.60%, precision was 79.58%, and F1 score was 79.16% in addition to accuracy. This minimizes overfitting by employing a sigmoid, Adam optimizer with a learning rate of 0.001 and a 0.2 dropout.

In order to optimize the DNN algorithm, a model variation with four hidden layers that uses an encoder-decoder architecture and applies a sigmoid activation function, an Adam optimizer with a learning rate of 0.001 and dropout of 0.2 to overcome overfitting, and other factors are found to perform better than the other variations.

TABLE 2
RESEARCH RESULTS USING THE DEEP NEURAL NETWORK ALGORITHM

Layers	Learning Rate	Accuracy	F1 Score	Precision	Recall	AUC
3 Hidden Decod-Encod	0.001	79.93%	81.53%	79.96%	79.70%	79.69%
3 Hidden Encod-Decod	0.001	79.41%	79.49%	79.67%	79.69%	79.70%
4 Hidden Decod-Encod	0.001	79.94%	80.63%	79.90%	80.01%	80.00%
4 Hidden Encod-Decod	0.001	79.24%	79.16%	79.58%	79.60%	79.56%
5 Hidden Decod-Encod	0.001	79.41%	80.53%	79.33%	79.40%	79.35%
5 Hidden Encod-Decod	0.001	79.67%	79.72%	79.94%	79.96%	79.95%
6 Hidden Decod-Encod	0.01	77.76%	75.39%	80.52%	78.66%	78.65%
6 Hidden Encod-Decod	0.001	77.93%	76.22%	79.83%	78.67%	78.68%
7 Hidden Decod-Encod	0.001	77.59%	77.21%	78.13%	77.99%	77.98%
7 Hidden Encod-Decod	0.001	78.02%	77.87%	78.42%	78.37%	78.36%
8 Hidden Decod-Encod	0.001	77.59%	76.76%	78.50%	78.10%	78.11%
8 Hidden Encod-Decod	0.001	78.02%	77.56%	78.65%	78.44%	78.47%

With this particular model variant, the DNN algorithm achieved 79.94% accuracy, 79.16% F1 score, 79.58% precision, 79.60% recall, and 79.56% AUC. When compared with other variations, the average difference in values between this variation and the others is 0.07% accuracy, 2.03% F1 score, 0.25% precision, 0.80% recall, and 0.90% AUC.

IV. Conclusion

General findings of the research topic, namely feature selection, are carried out after cleaning and selecting data used in the preprocessing stage. When applied to the Diabetic Retinopathy Debrecen dataset, the DNN data mining technique is an efficient algorithm that yields more accurate performance with a notable improvement in accuracy outcomes when compared to other folds. The research is conducted utilizing 10-fold cross-validation data. How precise the DNN algorithm with this model variation was able to reach 79.94%, while the F1 score reached 79.16%, precision of 79.58%, recall of 79.60%, and AUC of 79.56% with variations of four hidden layers encoder-decoder, sigmoid activation function, Adam optimizer, learning rate 0.001, and using dropout 02, which functions to prevent the occurrence of overfitting in the DNN process. When compared with other variations such as decoder-encoder, 3-8 hidden layers, learning rate 0.1 and 0.01, the average difference in values between this variation and the others is 0.07% accuracy, 2.03% F1 score, 0.25% precision, 0.80% recall, and 0.90% AUC.

DECLARATIONS

Conflict of Interest

The authors have declared that no competing interests exist.

CRediT Authorship Contribution

Cucu Ika Agustyaningrum: Formal analysis, Methodology, Data curation, Writing-Original draft, Writing-Reviewing and Editing, Visualization, Investigation; Haryani: Data curation, Writing-Reviewing and Editing, Investigation, Supervision, Funding Acquisition; Agus Junaidi: Writing-Reviewing and Editing, Investigation, Supervision, Funding Acquisition; Iwan Fadilah: Writing-Reviewing and Editing, Investigation.

Funding

Funding This paper is funded by the Bina Sarana Informatika Foundation, Republic of Indonesia, for the 2024 fiscal year (NO: 201/2.01/UBSI/VIII/2024, Title: Recipient of Internal Research and Community Service Grants Funding by the Bina Sarana Informatika Foundation in 2024).

Acknowledgment

Thanks are given to the Bina Sarana Informatika Foundation, which has provided research and community service assistance to lecturers, foundation/internal funds through the Research and Community Service Institute (LPPM) team to the Bina Sarana Informatika University Community, and lecturers and students who have helped. Thanks also to the master students of the Sepuluh November Institute of Technology (ITS) who have helped in this research collaboration.

REFERENCES

[1] T. O. Oladele, R. O. Ogundokun, A. A. Kayode, A. A. Adegun, and M. O. Adebiyi, "Application of data mining algorithms for feature selection and prediction of diabetic retinopathy", in *Proc. 2019 Int. Conf. Comput. Sci. Appl. Lecture Notes in Computer Science,* 2019, pp. 716–730, doi: 10.1007/978-3-030-24308-1_56.

[2] I. Odeh, M. Alkasassbeh, and M. Alauthman, "Diabetic retinopathy detection using ensemble machine learning," in *Proc. 2021 Int. Conf. Inf. Technol.*, 2021, pp. 173–178, doi: 10.1109/ICIT52682.2021.9491645.

[3] S. P. Dewi, N. Nurwati, and E. Rahayu, "Penerapan data mining untuk prediksi penjualan produk terlaris menggunakan metode k-nearest neighbor," *Build. Informatics, Technol. Sci.*, vol. 3, no. 4, pp. 639–648, Mar. 2022, doi: 10.47065/bits.v3i4.1408.

[4] I. Romli and A. T. Zy, "Penentuan jadwal overtime dengan klasifikasi data karyawan menggunakan algoritma c4.5," *Jurnal Sains Komput. Inform.*, vol. 4, no. 2, pp. 694–702, Sep. 2020, doi: 10.30645/j-sakti.v4i2.260.

[5] N. Gundluru *et al.*, "Enhancement of detection of diabetic retinopathy using Harris Hawks optimization with deep learning model," *Comput. Intell. Neurosci.*, vol. 2022, May 2022, Art. no. 8512469, doi: 10.1155/2022/8512469.

[6] D. H. Firdaus, B. Imran, L. D. Bakti, and E. Suryadi, "Klasifikasi penyakit katarak berdasarkan citra menggunakan metode convolutional neural network (CNN) berbasis web," *Jurnal Kecerdasan Buatan dan Teknol. Inf.*, vol. 1, no. 3, pp. 18–26, Dec. 2022, doi: 10.69916/jkbti.v1i3.6.

[7] M. Alizal, M. Susanto, A. Setyawan, H. Fitriawan, and M. Mardiana, "Sistem keamanan ruangan dengan human detection menggunakan sensor kamera berbasis deep learning," *Jurnal Teknoinfo*, vol. 18, no. 1, pp. 182–192, Jan. 2024, doi: 10.33365/jti.v18i1.3798.

[8] W. Andrian, "Klasifikasi diabetic retinopathy menggunakan arsitektur deep learning model CNN (convolutional neural network)," M. S. thesis, Univ. Teknol. Dig. Indonesia, Indonesia, 2023. [Online]. Available: https://eprints.utdi.ac.id/10007/

[9] J. F. Ath-Thaariq, R. A. Rajagede, and R. Rahmadi, "Kajian pustaka pengembangan aplikasi mobile untuk pengenalan bahasa isyarat berbasis deep learning," *Automata*, vol. 2, no. 1, 2021. [Online]. Available: https://journal.uii.ac.id/AUTOMATA/article/view/17360

[10] T. R. Gadekallu *et al.*, "Early detection of diabetic retinopathy using pca-firefly based deep learning model," *Electron.*, vol. 9, no. 2, Feb. 2020, Art. no. 274, doi: 10.3390/electronics9020274.

[11] T. R. Gadekallu, N. Khare, S. Bhattacharya, S. Singh, P. K. R. Maddikunta, and G. Srivastava, "Deep neural networks to predict diabetic retinopathy," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, pp. 5407–5420, May 2023, doi: 10.1007/s12652-020-01963-7.

[12] D. Das, S. K. Biswas, and S. Bandyopadhyay, "A critical review on diagnosis of diabetic retinopathy using machine learning and deep learning," *Multimed. Tools Appl.,* vol. 81, no. 18, pp. 25613–25655, 2022, doi: 10.1007/s11042-022-12642-4.

[13] D. S. O. Panggabean, E. Buulolo, and N. Silalahi, "Penerapan data mining untuk memprediksi pemesanan bibit pohon dengan regresi linear berganda," *Jurnal Ris. Komp*., vol. 7, no. 1, pp. 56–62, Feb. 2020, doi: 10.30865/jurikom.v7i1.1947.

[14] *Diabetic retinopathy Debrecen*, UC Irvine Machine Learning Repository, 2014. doi: 10.24432/C5XP4P.

[15] S. Watanabe and H. Nishimori. (2016). Fall lecture note on statistical learning theory. Lecture note for Tokyo Institute of Technology.

[16] D. N. Fathurrahman, A. B. Osmond, and R. E. Saputra, "Deep neural network for speech recognition on Sundanese language of the Middle East dialect (Majalengka)," *e-Proc. Eng.*, vol. 5, no. 3, pp. 6073–6080, Dec 2018. [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/7967

[17] C. I. Agustyaningrum, R. Dahlia, and O. Pahlevi, "Comparison of conventional machine learning and deep neural network algorithms in the prediction of monkey-pox," *J. Ris. Inform.* vol. 5, no. 3, pp. 253–262, Jun. 2023, doi: 10.34288/jri.v5i3.217.

[18] I. Elujide, S. G. Fashoto, B. Fashoto, E. Mbunge, S. O. Folorunso, and J. O. Olamijuwon, "Application of deep and machine learning techniques for multi-label classification performance on psychotic disorder diseases," *Inform. Med. Unlocked*, vol. 23, 2021, Art. no. 100545, doi: 10.1016/j.imu.2021.100545.

[19] M. Nabipour, P. Nayyeri, H. Jabani, S. S., and A. Mosavi, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," *IEEE Access*, vol. 8, pp. 150199–150212, Aug. 2020, doi: 10.1109/ACCESS.2020.3015966.

[20] A. C. Sitepu and M. Sigiro, "Analisis fungsi aktivasi ReLu dan Sigmoid menggunakan optimizer SGD dengan representasi MSE pada model backpropagation," *Jurnal Teknik Informatika Universal*, vol. 1, no. 1, pp. 12–25, 2021.

[21] M. Astiningrum, M. Mentari, and R. R. N. Rachma, "Deteksi kesegaran daging sapi berdasarkan ekstraksi fitur warna dan tekstur, *Seminar Informatika Aplikatif,* pp. 217–222, 2019.

[22] M. Syam, J. Raharjo, and R. Patmasari, "Identifikasi asal daerah berdasarkan suara manusia dengan metode linier predictive coding (LPC)," *e-Proc. Eng.*, vol. 6, no. 3, pp. 10226–10233, Dec. 2019. [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/11346

[23] H. Abhirawa, J. Jondri, and A. Arifianto, "Pengenalan wajah

menggunakan convolutional neural network," *eProc. Eng.*, vol. 4, no. 3, pp. 4907–4916, 2017. [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/5420

[24] C. D. Suhendra and A. C. Saputra, "Penentuan parameter learning rate selama pembelajaran jaringan syaraf tiruan backpropagation menggunakan algoritma genetika," *Jurnal Teknologi Informasi Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika,* vol. 14, no. 2, pp. 202–212, 2020, doi: 10.47111/jti.v14i2.1141.

[25] Y. Wang, J. Liu, J. Misic, V. B. Misic, S. Lv, and X. Chang, "Assessing optimizer impact on DNN model sensitivity to adversarial examples," *IEEE Access*, vol. 7, pp. 152767–152776, 2019, doi: 10.1109/ACCESS.2019.2948658.

[26] N. A. Putro, R. Septian, W. Widiastuti, M. Maulidah, and H. F. Pardede, "Prediction of hotel booking cancellation using deep neural network and logistic regression algorithm," *Techno Nusa Mandiri : J. Comput. Inform. Technol.*, vol. 18, no. 1, pp. 1–8, 2021, doi: 10.33480/techno.v18i1.2056.

[27] S. Mandt, M. D. Hoffman, and D. M. Blei, "Stochastic gradient descent as approximate Bayesian inference," *J. Mach. Learn. Res.*, vol. 18, 2017, Art. no. 134, [Online]. Available: https://www.jmlr.org/papers/v18/17-214.html

[28] M. W. P. Aldi, J. Jondri, and A. Aditsania, "Analisis dan implementasi long short term memory neural network untuk prediksi harga bitcoin," *e-Proc. Eng.*, vol. 5, no. 2, pp. 3548–3555, Aug. 2018 [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/6739

[29] Y. Christian, "Comparison of machine learning algorithms using weka and sci-kit learn in classifying online shopper intention," *J. Inform. Telecommun. Eng.* vol. 3, no. 1, pp. 58–66, 2019, doi: 10.31289/jite.v3i1.2599.

[30] L. Ratnawati and D. R. Sulistyaningrum, "Penerapan random forest untuk mengukur tingkat keparahan penyakit pada daun apel," *Jurnal Sains Seni ITS*, vol. 8, no. 2, pp. A71–A77, 2019, doi: 10.12962/j23373520.v8i2.48517.

[31] M. A. A. Sarofi, I. Irhamah, and A. Mukarromah, "Identifikasi genre musik dengan menggunakan metode random forest," *Jurnal Sains dan Seni ITS*, vol. 9, no. 1, pp. D79–D86, 2020, doi: 10.12962/j23373520.v9i1.51311.

[32] C. I. Agustyaningrum, M. Haris, R. Aryanti, and T. Misriati, "Online shopper intention analysis using conventional machine learning and deep neural network classification algorithm," *Jurnal Penelitian Pos Informatika*, vol. 11, no. 1, pp. 89–100, 2021, doi: 10.17933/jppi.v11i1.341.

[33] S. I. Gultom, "Implementasi data mining menentukan pola hidup sehat bagi pengguna KB menggunakan algoritma adaboost (studi kasus :dinas Serdang Bedagai)," *Jurnal Informasi Teknologi Ilmiah*, vol. 7, no. 3, pp. 298–304, Jun. 2020. [Online]. Available: https://ejurnal.stmik-budidarma.ac.id/index.php/inti/article/view/2405